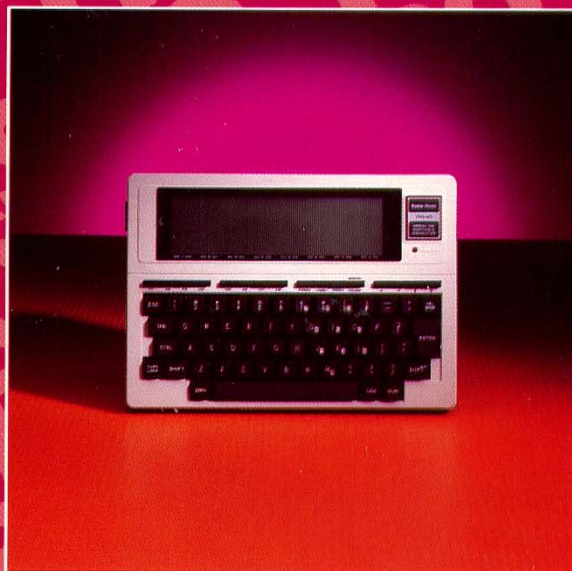


The TRS-80 User's Encyclopedia (Model 100)

Gary Phillips, Jacquelyn Smith and Julia Menapace

Includes
NEC
PC 8201



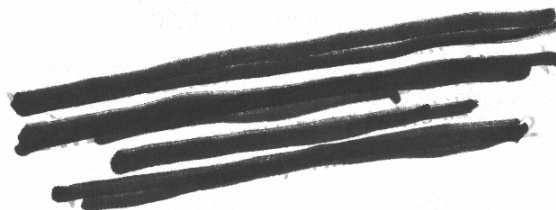
ARRAYS, INC.
THE BOOK DIVISION



THE TRS-80 USER'S ENCYCLOPEDIA

(Model 100 and NEC PC 8201)

by
**Gary Phillips
Jacquelyn Smith
Julia Menapace**



ARRAYS, INC.

ARRAYS, INC./THE BOOK DIVISION
11223 S. Hindry Avenue • Los Angeles, CA 90045

ACKNOWLEDGEMENTS

Staff for *The TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201)*:

Arrays Inc./The Book Division:

Michael Mellin
Executive Editor

Robert A. Sandberg
Editor-in-Chief

Editor/Proofreaders
Mia McCroskey
Mark Bodziak

Editorial Assistant
Jennifer Boynton

Production Editor
Robert A. Sandberg

Production
Estela Montesinos
Argelia Navarrete
Steve Gunn
Mark Mansell
Virginia Adams
Mike Hemby

Gary Phillips and Associates:

Project Management
Jacquelyn Smith

Senior Technical Editors
Robert F. Arbegast
Scott K. Razak

Technical Editors
Christopher Horn
Steven P. Kennedy
Robert Knedlik
Darlene P. Mortenson
Robin Vidaurri
Christopher Wirth

General Assistant
Emil Flock

Julia C. Menapace wishes to acknowledge Joseph Linus for his patience and support and for periodically liberating me to the beach. Some of the general microcomputer definitions in the book have been adapted or reprinted by permission from the *International Microcomputer Dictionary* published by SYBEX, Inc., copyright 1981, SYBEX, Inc.

Other general definitions have been reprinted from *The Reference Encyclopedia for the IBM Personal Computer*, published by Ashton-Tate, copyright 1983, Ashton-Tate.

ISBN 0-912003-13-8

Copyright 1984 © Arrays Incorporated/The Book Division. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Arrays Inc./The Book Division.

TRS-80 is a registered trademark of the Radio Shack Division of Tandy Corporation. The use of trademarks or other designations is for reference purposes only.

PREFACE

The *TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201)* is your one definitive reference book for the Model 100 and NEC PC 8201 Computers. By deciding to own it, you have made an investment in your computing future. It will save you hundreds of hours, by bringing material from many sources into a single, easy-to-use, alphabetical reference handbook.

Much of the material is not available anywhere else. Here you will find the essential background material you need to answer your questions and solve problems on the Model 100 or NEC PC 8201 Computer.

Your *User's Encyclopedia* guides you through machine operation, BASIC programming, and what's available in software and hardware. Your *Encyclopedia* will quickly pay for itself through product information alone. It is a comprehensive source of information about the whole area of microcomputers. The entries are short, clear, self-contained, and understandable. Should you desire additional information, hundreds of cross-references point you to related entries in the *Encyclopedia*. You will find it one of the most frequently used and most highly treasured books in your personal computer library. Many who buy the *Encyclopedia* come to swear by it.

The *Encyclopedia* includes the BASIC program listings for several useful utilities such as determining the size of files, the amount of free RAM, and showing selected parts of RAM on the screen or printer.

While your *User's Encyclopedia* was written in simple language so beginning users can understand it, you will continue to find it indispensable as your knowledge of computers grows. While it does not offer a detailed coverage of the internal electronic details of the computers, it does provide a complete reference on BASIC programming, general operation of the computer and its accessories, and available products.

You will usually use your *TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201)* as a quick reference source. You will probably keep it near your computer, instantly available when you encounter difficulties or want a more complete understanding of what your Model 100 or NEC PC 8201 computer is doing. You will also enjoy just browsing through its many short, easy-to-read-and-understand entries. This allows you to easily and casually increase your general knowledge of the Model 100 and NEC PC 8201 Computers, and microcomputers in general. And now that you own it, you will probably find still other uses for the most universal, valuable, and easy-to-use book available for the Model 100 and NEC PC-8201.

The *TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201)* was produced for you by Gary Phillips & Associates. Gary Phillips & Associates also provides *User's Encyclopedias* for the IBM PC, Atari, Apple, Timex/Sinclair, Commodore-64, TI-99, and many other personal computers. The *TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201)* will serve as your first reference source for most questions. Manuals from Radio Shack or NEC and other books provide a valuable backup

to the *User's Encyclopedia*—when you have time to explore a topic in more formal book presentation. But when time is pressing and you want information fast, you can rely on the *TRS-80 User's Encyclopedia (Model 100, and NEC PC 8201)*.

The *TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201)* is organized alphabetically, with numbers and special symbols following "Z". Here are some special pointers for understanding the format of the *Encyclopedia*:

Variable types in various statements and commands are placed between arrow brackets. For example, <line number> means you would use a specific line number from your BASIC program. Similarly, <variable> indicates any variable you are using in your program. For example, the statement to display a variable's value in PRINT <variable>. To print out the contents of the variable Count, you would replace <variable> with Count, giving PRINT Count. Parentheses () around an item in a command or instruction should be typed by the user at the keyboard. However, brackets [] indicate that the enclosed material is optional; these brackets should not be entered into the computer. Also, q.v. is used after a word or phrase to direct you to other entries that you may want to see for additional information (q.v. from the Latin quo vide, "which see"). The names and addresses of all software and hardware manufacturers are listed alphabetically in an appendix.

NEC PC 8201 user's should scan the articles under NEC PC 8201 for a concise summary of the differences between the PC 8201 and the Model 100. The material in this book applies to both machines, but the differences between the machines must be understood and kept in mind when reading any article.

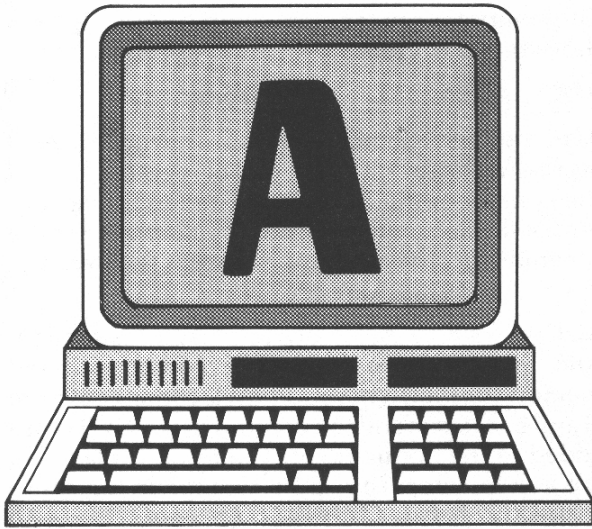
We have made every effort to make your *TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201)* complete, accurate, and up-to-date. The descriptions of products are identified by an "*" following the product name. These may be either reviews or brief descriptions to help you find products you may want and acquire further information. You will need to check with the manufacturer or a retail outlet to verify the suitability of the product for your needs and to determine its price and availability. Similarly, you should verify any technical information in the *Encyclopedia* before relying on it for a major decision. Neither the authors, Gary Phillips & Associates, nor Arrays, Inc./The Book Division will be liable for any errors or omissions in the *User's Encyclopedia*. If you should find an error or omission, or have any suggestions for improvements or additions to future printings or revisions of the *TRS-80 User's Encyclopedia (Model 100, and NEC PC 8201)*, please write us:

Arrays, Inc./The Book Division
TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201)
294 Donahue Street
Sausalito, California 94965

Manufacturers of hardware or software who would like to be included in the next revision of the *TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201)* may send review copies and descriptive literature to the same address.

Congratulations on deciding to own the one definitive reference source for the Model 100 and NEC PC 8201 Computers. We know you will enjoy and profit from it in the months to come.

Gary Phillips
Jacquelyn Smith
Julia C. Menapace
Staff of Gary Phillips & Associates



A Codes. ASCII 65, HEX 41. a—ASCII 97, HEX 61.

A Diagram symbol meaning Accumulator. Also stands for Address line, and the hexadecimal sign for the decimal integer 10: A base 16 = 10 base 10 = 12 base 8 = 1010 base 2

Å Symbol for angstrom, equalling one billionth of a meter (one nanometer).

Abort The process of manually ending a program or process which has encountered an error condition and returning control to an operator or operating system. For most applications, this is accomplished by pressing the BREAK/PAUSE and SHIFT keys simultaneously.

ABS BASIC Function. ABS disregards negative signs to return the positive, or absolute value of the expression <numvar>. The format is:

<variable> = ABS(<numvar>)

<numvar> may be any numeric expression. For example:

```
Ok
PRINT ABS(-100)
100
Ok
```

The positive (absolute) value of -100 is 100.

Absolute Value The value of a number expressed as a positive number. Denoted by enclosure in vertical bars (|). Thus, if a number is positive or zero $|x| = x$. If x is negative: $|x| = -x$. See ABS.

A-Bus The primary internal source-bus to the arithmetic logical unit in any processor.

AC Alternating electrical Current.

AC or ACC ACCumulator, or ACCess time: the time it requires to get a byte of data from memory.

Accounting Software See Business Manager Series, The; BusinessPak+; Travelling Accountant, The; Travelling Expense Manager, The; EXPNS+; and PortaFin.

Accumulator A register into which the results of arithmetic operations are stored. More than one accumulator can be present in a central processor. The 80C85 CPU in the Model 100 has six general purpose registers, GPR's, which often function as accumulators.

AC/DC Adapter Unit The Model 100 gets its operational power via the Radio Shack AC Power Supply (RS-26-3804), from one of two sources—four AA alkaline batteries or wall current. The adapter converts standard 120V AC to 6V DC which is the type of current the Model 100 uses. To use the power adapter, simply plug the two prong plug into a wall outlet and the much smaller one prong plug into the jack labeled DC 6V on the right side panel of the Model 100. It is good practice to make this connection while the computer is turned off. Use of either the power adapter unit or AA batteries recharges the built-in ni-cad (nickel-cadmium) battery that maintains RAM memory while the computer is turned off.

ACI 80C85 Assembly Language Instruction. Add immediate with carry. The content of the second byte of the instruction and the content of the CY flag are added to the contents of the accumulator. The result is placed in the accumulator. The addressing mode is immediate. Z, S, P, CY and AC flags are set.

ACIA Asynchronous Communication Interface Adapter.

ACK ACKnowledge character in ASCII, a 06 base 16. This is used in communications to complete a handshaking sequence. The ACK signal indicates that the information has been accepted.

ACM Association for Computing Machinery. The major international society for computer technology is the ACM. Through its numerous publications and special interest groups, ACM will be of interest to many Model 100 users. ACM has special interest groups that discuss many topics. For more information write to ACM or find the *Journal of the ACM* at your library.

Association for Computing Machinery
11 W. 42nd St. 3rd Floor
New York, NY 10036
(212) 869-7440

Acoustic Coupler • Address

Acoustic Coupler A mechanical instrument for connecting the telephone handset, through a modem, to a computer. The data is converted to tones—usually audible—for transmission over the phone lines.

The Model 100's built-in modem can connect the computer to another computer or information service over the phone lines using one of two optional devices. They are the Model 100 Modem Cable (RS-26-1410) or the Model 100 Acoustic Coupler (RS-26-3805). The acoustic coupler, while more prone to "noise" than the direct connect modem cable, has the advantage of being usable on phones without a modular jack, such as those in phone booths or in many motel rooms. However, the acoustic coupler does not support the autodial and autolog-on functions of TELCOM.

To use the acoustic coupler with the Model 100, set the DIR/ACP and ANS/ORIG switches on the left side panel of the computer to ACP and ORIG respectively. This places the computer in originate mode, ready to call another computer or host system such as Compuserve or The Source, through the acoustic coupler.

Now ENTER TELCOM mode. You will automatically be in Entry mode with the default communications parameters already set. They will be correct for most of the larger commercial services, but should you want to change them you may use F3, the STAT key, in TELCOM Entry mode. This displays the current communications parameters and allows you to change them. Note that if you change the default baud rate, the built-in modem is disabled. Next, manually dial the number for the host computer on the telephone. When you hear a high-pitched tone (the carrier frequency) over the ear piece, place the receiver cup of the coupler over the ear piece of the handset and press F4, the TERM key, to place the computer in Terminal mode. Notice that the function key definitions are different in Terminal mode. You are now ready for direct computer to computer communications. See TELCOM, Communications Parameters—How to Set.

ACT ACcumulator, Temporary (in the 80C85 microprocessor). In the Model 100, one input to the arithmetic logic unit (ALU) comes from the main accumulator, and the second from the temporary register, register B. Registers W and Z are also temporary registers, but are not accessible to the programmer.

A/D "A to D" or Analog to Digital. Conversion from a sensor's analog voltages and currents to the digital representation used by computer systems.

The computer can then process data directly from the external world.

ADC Analog to Digital Converter.

ADC 80C85 Assembly Language Instruction. ADD register with Carry. The content of register <r> and the content of the carry bit are added to the content of the accumulator. The result is placed in the accumulator. The addressing mode is register. Z, S, P, CY, and AC flags are set.

ADC M 80C85 Assembly Language Instruction. Add memory with carry. The contents of the memory location whose address is contained in the H and L registers and the content of the CY flag are added to the accumulator. The result is placed in the accumulator. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

ADCCP Advanced Data Communication Control Procedures.

ADD 80C85 Assembly Language Instruction. ADD Register. The content of register <r> is added to the content of the accumulator. The result is placed in the accumulator. The addressing mode is register. Z, S, P, CY, and AC flags are set.

Add Characters in Between Existing Characters

This is done in TEXT mode as invoked by the BASIC EDIT command, by using the cursor movement keys to position the cursor within the existing text, or program lines, at the point where you wish to insert new characters. Then, as you key, the characters to the right of the cursor move right one column for each new character that is inserted, automatically wrapping around to the next line when the screen width of forty characters is exceeded. See EDIT.

ADD M 80C85 Assembly Language Instruction. ADD Memory. The content of the memory location whose address is contained in the H and L registers is added to the content of the accumulator. The result is placed in the accumulator. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

Adder A processor unit that performs binary arithmetic.

Add-On System or Circuitry attached to a computer to increase memory or performance.

Address Position of a word in memory, expressed by a number. In the Model 100 addresses range from 0 to 65,535. RAM memory-address ranges depend upon how many K your model has. See Address Notation.

RAM Size	Memory Address Range
8K	57344 - 62960
16K	49152 - 62960
24K	40960 - 62960
32K	32768 - 62960

Note that in each case the highest RAM memory address is 62960, the value of BASIC system constant MAXRAM.

Address Mark 8 bit code placed on diskette track at the beginning of specific fields on a diskette track, such as the index, identification, data, or deleted data fields.

Address Notation Addresses for the Model 100 may be written in any of three formats: decimal, hexadecimal, or LSB, MSB. Decimal format simply specifies the sequential number of a byte in decimal (with numbering starting from zero). The first byte in memory has address 0, the second has address 1, etc. The hexadecimal format has the same system, but with the number of the byte given in the hexadecimal (base 16) number system. Thus, the byte with decimal address 65024 (start of screen image) may also be referred to as hexadecimal byte 2621, FEE) 65024 base 10 = FEE0 base 16.

LSB, MSB format specifies an address in the internal format of the 80C85 processor. Since one byte can only store a number in the range of 0-255, Model 100 addresses require two bytes.

Each byte in the memory can be addressed by a four-digit hexadecimal number C0-ffff, which corresponds to 0-65535 in decimal notation. Since two hex digits can be stored in one byte, the first two hex digits (the most significant) are stored in one byte, the last two (the least significant) are stored in another. Now, for the catch. In order to minimize the amount of internal work the 80C85 has to do to manipulate two-byte addresses, they are stored in reverse order—the least significant byte (LSB) first, then the most significant byte (MSB). This form of address storage is accordingly called LSB, MSB format. Address FEE0 stored in LSB, MSB format would have E0 in the first byte (the LSB) followed by FE (the MSB). This may also be written as EO,FE. It would appear in a dump as E0FE. Remember that the pairs of hexadecimal digits are reversed, and the actual address is FEE0. See Address.

Address of Variable in Memory BASIC. See VARPTR.

Address Organizer See ADDRSS Mode.

ADDRSS Mode The Model 100 address organizer program provides a means for locating and retrieving phone numbers, addresses, and other address related information. In order to use it, you must create a file in text mode named ADRS, containing the address information. Each address entry in the file should constitute one paragraph, that is, it should be terminated by a carriage return. When the TEXT program saves the file to RAM, it automatically gives it the extension .DO, signifying that it is a text file. The ADDRSS program then operates upon that file to find and display all of the occurrences of the string you input for the program to find. If you try to enter the ADDRSS program without first creating an ADRS.DO file, the computer beeps and prints the message:

```
ADRS.DO not found
Press space bar for MENU
```

The ADRS.DO file you create using the TEXT processor may also be used by the TELCOM program to autodial phone numbers and autolog onto information services such as Compuserve and The Source. Although the ADDRSS program places no restrictions on how you organize the records or entries in the ADRS.DO file, if you want to use the file with the TELCOM program as well there are some restrictions. The TELCOM program requires that the ADRS.DO file address records be organized with the name, then the phone number (bracketed by colons and with the segments optionally separated by dashes), and finally the address last.

To enter the ADDRSS program, position the cursor on the Main Menu over ADDRSS (the name of the address organizer applications program) and press <ENTER>. You will then see the prompt ADRS to let you know you are in the ADDRSS mode and the eighth screen line will display the current function key definitions. The ADDRSS program uses three of the six programmable function keys. They are:

F1—Finds and displays on the LCD all address records containing the string supplied by the user. If no string is given, F1 displays the entire contents of the ADRS.DO file on the LCD. If there are more records than will fit on the LCD screen, F3 and F4 are enabled as <MORE> and <QUIT> keys. Press F3 to see more records. Press F4 to exit.

F5—Finds and prints on the printer all address records containing the string supplied by the user. If a null string is entered F5 prints the entire contents of the ADRS.DO file on the printer.

F8—Exits the ADDRSS program and returns to the Main Menu.

Adhesive Tab • Amplifier

Pressing F1 or F5 displays this prompt on the LCD screen:

ADRS:FIND

Key in the string you wish to locate and press the ENTER key. It may be of any length, and generally represents any portion of the address you can remember, such as a name, town, zip code, etc. The ADDRSS program treats upper and lower case the same, and a search string is considered found when it is located, even as part of a longer word.

To view or print all the address records in the ADRS.DO file, press F1 or F5 to enable the find function, but do not specify a string to find. Instead just hit the ENTER key. The program will then list all records to the LCD or the printer in the same way it lists selected records when you specify a find string.

Adhesive Tab A tab used to cover the write-protect notch on floppy disks. See Write-Protected Diskette.

ADI 80C85 Assembly Language Instruction. ADd Immediate. The content of the second byte of the instruction is added to the content of the accumulator. The result is placed in the accumulator. The addressing mode is immediate. Z, S, P, CY, and AC flags are set.

ADRS.DO File See ADDRSS Mode, TELCOM Mode.

Advance One Line on Printer To space up one line on the Epson RX-80 printer, enter the BASIC statement:

LPRINT CHR\$(10)

or use the "line feed" (LF) button of the printer.

Just LPRINT gives a line feed—both space up one line (line feed) and return to left margin (carriage return).

Advance to Top of Page To cause a form feed on the Epson RX-80 printer, enter the BASIC statement:

LPRINT CHR\$(12)

Or use the "top of form" or "form feed" manual control button (FF) on the printer.

You may then need to adjust the paper in the printer so it actually is at the top of a page as defined by the perforations.

In a program, you should provide instructions to the operator and a pause (q.v.) to allow for adjustment of the paper.

Advance Word In TEXT mode, or TEXT as invoked by the BASIC EDIT command, you may cause the cursor to advance one word at a time. Pressing SHIFT and → at the same time advances the cursor from its current position to the first character of the next word to the right.

Alexis Adventure* Voyage across the oceans in a search for your lost kingdom. Many perils await you throughout your adventure. 24K; cassette. SilverWare.

Algorithm A solution to a problem with step-by-step specifications, ending in a finite time. You state a problem, develop an algorithm for its solution, flowchart the solution steps, and then develop a program from the flowchart.

Allocation of Space Assigning particular areas of memory (internal RAM, cassette, etc.) to particular files, programs, or functions. A programmer or program sometimes does the allocation.

Alpha and Beta Test Sites A test site participates with the originators of a hardware or software product to test it in a real world situation.

Alpha test usually involves only a very few companies or individuals who realize that the product is incomplete or may have flaws. Alpha testers often are internal to the originating company or are otherwise closely related.

Beta test sites are generally larger in number. They expect that the product is essentially complete and correct, and agree to use it in a real world production situation. If errors are discovered, the originators ordinarily attempt to fix them rapidly so the Beta sites can stay "on the air" with the new product. If numerous or serious bugs are found the product may have to go back to alpha testing until an improved version can be presented for another round of Beta testing.

Alphanumeric A set composed of alphabetic and numeric characters.

Alterable Memory Storage media, such as a RAM diskette or cassette which can be written onto or changed.

Alternating Current Any signal not constant is considered alternating current. However, this term usually means that the current regularly changes polarity. Although the Model 100 uses 6V DC (direct current), it will also run on U.S. standard 120 V AC wall current with the help of the Radio Shack AC Power Supply Unit (RS-26-3804). See AC/DC Adapter Unit.

ALU See Arithmetic Logic Unit.

Ampere Measurement of electrical current: the actual number of electrons moving past a stated point per second.

Amplifier A device or circuit that increases the power or strength of a signal.

ANA 80C85 Assembly Language Instruction. AND register. The content of register <r> is logically ANDed with the content of the accumulator. The result is placed in the accumulator. The CY flag is cleared. The addressing mode is register. Z, S, P, CY, and AC flags are set.

ANA M 80C85 Assembly Language Instruction. AND memory. The contents of the memory location, whose address is contained in the H and L registers, is logically ANDed with the content of the accumulator. The result is placed into the accumulator. The CY flag is cleared. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

Analog Having a continuous range of voltage or current values. See Digital, A/D.

Analyser Or, Analyzer. Any device that checks or regulates a component, board, or system and presents the data for review.

AND Term for a logical procedure defined by the rule: if A AND B are 1, then C is 1, otherwise C is 0. The AND of 10110111 and 10000100 is 10000100. For logical situations such as the conditional test in an IF statement, substitute TRUE for 1 and FALSE for 0.

ANI 80C85 Assembler Language Instruction. AND Immediate. The content of the second byte of the instruction is logically ANDed with the contents of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared. The addressing mode is immediate. Z, S, P, CY, and AC flags are set.

ANSI American National Standards Institute.

APL A Programming Language, invented by Kenneth Iverson. It is used for algorithmic interactive programming.

Apostrophe (') Use for comments in BASIC. A synonym for REM used to include remarks or explanatory comments in a program. The colon (:) allows multiple BASIC statements on one line. The apostrophe can be used without a colon to indicate that the rest of the line is a remark. Examples:

```
20 REM Just a comment
30 ' Just a Comment
40 LET X = 1: REM Just a comment
50 LET X = 1: ' Just a comment
60 LET X = 1' Just a comment
```

Append To add to the end of a character string or list.

Applications Software A software package is a group of computer programs, possibly including data files and documentation, which perform a

function or group of related functions on the computer. These are called applications software when the programs are used by the operator, such as a word processing package or an accounting package. These programs are called systems software packages when they facilitate the use of the machinery, such as a data base management package, a disk operating system, or a program development package.

Arbitration Management of claims by competing systems or processes for a limited resource. Bus arbitration allocates a system bus among the subsystem components, i.e. the CPU memory, disk controller, and other external devices.

Architecture The special selection, design, and inter-connection of the principal components of a system. In a microprocessing unit this could be the number and function of registers, the instruction addressing modes, and the bus structure and timing.

Arctangent BASIC. See ATN.

Argument Data passed from one process or program to another process or program. Similar to a football pass, except that one or more bytes of data replaces the football. Also, the receiver is a program. The sender may be a program or a person typing the data onto a command line to be "passed" to a program. The most common example would be a BASIC program passing variables to a subroutine. See CALL.

Arithmetic Logic Unit ALU. The element which performs the basic data manipulations in the central processor like add, subtract, complement, negate, rotate, AND and OR.

Arithmetic Statement An instruction specifying an arithmetic operation. See BASIC Math Functions, BASIC Math Operators.

Armchair Quarterback Strategy* Outpro the pros. 16K. Chattanooga Choo Choo Software.

ARQ Automatic ReQuest for repeat. In telecommunications, a device capable of determining whether it has correctly received information transmitted from another source may automatically request a repeat transmission.

Arrays BASIC. See DIM.

Arrow Down (↓) Pressing the down arrow moves the cursor down one line in the same column position on the screen it previously occupied. This function is common in text and command entry. Holding down the ↓ key causes this process to auto

Arrow Left • ASCII—Saving BASIC Program

repeat, scrolling down the current column through the text.

In TEXT mode, and in TEXT as invoked by the BASIC EDIT command, the ↓ key performs additional functions if used in conjunction with the SHIFT or CTRL key. Pressing SHIFT and ↓ advances the cursor to the last line in the same column. If the cursor is already on the last line and you press this key combination, the next seven lines of the current file scroll onto the LCD. Meanwhile what was the eighth line of text moves up to the first line of the LCD with the cursor remaining in the same place within that line. Pressing CTRL and ↓ together takes you to the last eight lines of the file, displaying those lines on the LCD and placing the cursor after the last character.

Arrow Left (←) Pressing the left arrow moves the cursor left one character. This function is common in text and command entry. Holding down the ← key causes this process to auto repeat, scrolling left along the current line. In some applications, when the cursor reaches the left-most position of the current line, it wraps around to the right-most position of the previous line to repeat the process. As long as the ← key is held down, the cursor continues to move to the left and in the direction of the beginning of the file. In other cases the cursor left movement is confined to the current line. In all modes, except TEXT and BASIC EDIT backspacing, the cursor with the ← key erases the characters the cursor passes over.

Arrow Right (→) Pressing the right arrow moves the cursor right one character. This function is common in text and command entry. Holding down the → key causes this process to auto repeat, scrolling right along the current line. When the cursor reaches the right-most position of the current line, it wraps around to the left-most position of the next line to repeat the process. As long as the → key is held down, the cursor continues to move to the right and in the direction of the end of the file. In TEXT, and in TEXT as invoked by the BASIC EDIT command, the → key performs additional functions if used in conjunction with the SHIFT or CTRL key. Pressing SHIFT and → moves the cursor to the first character of the next word to the right. Pressing CTRL and → moves the cursor to the right-most character on the current line.

Arrow Up (↑) Pressing the up arrow moves the cursor up one line in the same column position on the screen it previously occupied. This function is common in text and command entry. Holding down the ↑ key will cause this process to auto

repeat scrolling up the current column through the text.

In TEXT, and in TEXT as invoked by the BASIC EDIT command, the ↑ key performs additional functions if used in conjunction with the SHIFT or CTRL key. Pressing SHIFT and ↑ moves the cursor to the first line, same column. If the cursor is already on the first line and you press this key combination, the previous seven lines of the file scroll onto the LCD. Meanwhile what was the first line of text moves down to eighth line of the LCD with the cursor remaining in the same place within that line.

ASC BASIC Function. ASC returns the ASCII code for the first character of the string <strvar>. The format is:

<var> = ASC(<strvar>)

<strvar> may be any string expression. For example:

```
Ok
10 X$ = "SAMPLE"
20 PRINT ASC(X$)
RUN
83
Ok
```

In this example, the ASCII code for "S" is 83. "S" is the first character of the string "SAMPLE". If x\$ is undefined, BASIC displays the message, ?FC ERROR, meaning that an illegal function call has occurred. For a listing of the ASCII codes see ASCII Character Code Table.

ASCII American Standard Code for Information Interchange. This is a standardized seven-bit code of numeric equivalents for characters. Decimal 0-127 are seven-bit ASCII codes. Many computers also use the seven-bit extension of the ASCII character codes to represent special graphics characters. Decimal 128-255 are 8-bit ASCII codes. On the Model 100 all 256 ASCII codes may be generated on the keyboard by pressing one or a combination of two of these keys. All ASCII characters except 0-32 display on the LCD. These first 33 invisible ASCII codes are known as control codes and perform special functions on the computer and printer, as well as in some software applications. See ASCII Character Code Table, CODE Key, GRPH Key, CHR\$, Control Characters—Computer, Control Characters—Printer, Control Codes Text.

ASCII—Saving BASIC Program To save a program in ASCII rather than compressed format, allowing you to MERGE two programs together, put ,A after the close quote of the device: filename, or configuration sequence:

```
SAVE "RAM:SAMPLE.DO",A
```

ASCII—Saving Text File • ASCII Character Code Table

See MERGE.

To SAVE a program named "SAMPLE" in ASCII format to RAM, enter:

SAVE "SAMPLE.DO",A

To save the file on cassette, first rewind the cassette, press PLAY and RECORD together, then enter:

SAVE "CAS:SAMPLE",A

If you specify COM:, MDM:, LCD:, or LPT: as the device in your SAVE instruction, the file will automatically be transmitted in ASCII format.

The SAVE instruction does not alter your program in memory. It is important to be aware that if you write a BASIC program, it will be lost (erased) unless you SAVE it before you LOAD a new program to BASIC or use the NEW command.

To run the program at a later time, use the LOAD

command to copy it from the device you saved it on back into BASIC. See LOAD.

ASCII—Saving Text File Although your text file will be automatically saved as a RAM file when you exit TEXT by pressing F8, the MENU key or the ESC key twice, you may also make a copy off-line on cassette tape. You can do this while still in TEXT by pressing F3, the SAVE key. The prompt "Save to" will appear in the lower left corner of the screen, at which point you may key in a six character name for your file. Make sure that the cassette recorder is properly connected and the tape is in position ready to record. Pressing ENTER begins the recording process. Listen for a high pitched sound and make sure the tape moves. When the copy is made, the "save to" prompt disappears.

ASCII Character Code Table

Decimal	Hex	Binary	Printed Character	Keyboard Character
0	00	00000000		PAUSE
1	01	00000001		CTRL A
2	02	00000010		CTRL B
3	03	00000011		CTRL C
4	04	00000100		CTRL D
5	05	00000101		CTRL E
6	06	00000110		CTRL F
7	07	00000111		CTRL G
8	08	00001000		CTRL H
9	09	00001001		CTRL I
10	0A	00001010		CTRL J
11	0B	00001011		CTRL K
12	0C	00001100		CTRL L
13	0D	00001101		CTRL M
14	0E	00001110		CTRL N
15	0F	00001111		CTRL O
16	10	00010000		CTRL P
17	11	00010001		CTRL Q
18	12	00010010		CTRL R
19	13	00010011		CTRL S
20	14	00010100		CTRL T
21	15	00010101		CTRL U
22	16	00010110		CTRL V

ASCII Character Code Table

Decimal	Hex	Binary	Printed Character	Keyboard Character
23	17	00010111		CTRL W
24	18	00011000		CTRL X
25	19	00011001		CTRL Y
26	1A	00011010		CTRL Z
27	1B	00011011		ESC
28	1C	00011100		→
29	1D	00011101		←
30	1E	00011110		↑
31	1F	00011111		↓
32	20	00100000		SPACEBAR
33	21	00100001	!	!
34	22	00100010	”	”
35	23	00100011	#	#
36	24	00100100	\$	\$
37	25	00100101	%	%
38	26	00100110	&	&
39	27	00100111	,	,
40	28	00101000	((
41	29	00101001))
42	2A	00101010	*	*
43	2B	00101011	+	+
44	2C	00101100	,	,
45	2D	00101101	-	-
46	2E	00101110	.	.
47	2F	00101111	/	/
48	30	00110000	0	0
49	31	00110001	1	1
50	32	00110010	2	2
51	33	00110011	3	3
52	34	00110100	4	4
53	35	00110101	5	5
54	36	00110110	6	6
55	37	00110111	7	7
56	38	00111000	8	8
57	39	00111001	9	9

ASCII Character Code Table

Decimal	Hex	Binary	Printed Character	Keyboard Character
58	3A	00111010	:	:
59	3B	00111011	;	;
60	3C	00111100	<	<
61	3D	00111101	=	=
62	3E	00111110	>	>
63	3F	00111111	?	?
64	40	01000000	@	@
65	41	01000001	A	SHIFT A
66	42	01000010	B	SHIFT B
67	43	01000011	C	SHIFT C
68	44	01000100	D	SHIFT D
69	45	01000101	E	SHIFT E
70	46	01000110	F	SHIFT F
71	47	01000111	G	SHIFT G
72	48	01001000	H	SHIFT H
73	49	01001001	I	SHIFT I
74	4A	01001010	J	SHIFT J
75	4B	01001011	K	SHIFT K
76	4C	01001100	L	SHIFT L
77	4D	01001101	M	SHIFT M
78	4E	01001110	N	SHIFT N
79	4F	01001111	O	SHIFT O
80	50	01010000	P	SHIFT P
81	51	01010001	Q	SHIFT Q
82	52	01010010	R	SHIFT R
83	53	01010011	S	SHIFT S
84	54	01010100	T	SHIFT T
85	55	01010101	U	SHIFT U
86	56	01010110	V	SHIFT V
87	57	01010111	W	SHIFT W
88	58	01011000	X	SHIFT X
89	59	01011001	Y	SHIFT Y
90	5A	01011010	Z	SHIFT Z
91	5B	01011011	[[
92	5C	01011100	\	GRPH -

ASCII Code to Characters • ASCII Control Characters

Decimal	Hex	Binary	Printed Character	Keyboard Character
93	5D	01011101]]
94	5E	01011110	^	^
95	5F	01011111	—	—
96	60	01100000	\	GRPH [
97	61	01100001	a	A
98	62	01100010	b	B
99	63	01100011	c	C
100	64	01100100	d	D
101	65	01100101	e	E
102	66	01100110	f	F
103	67	01100111	g	G
104	68	01101000	h	H
105	69	01101001	i	I
106	6A	01101010	j	J
107	6B	01101011	k	K
108	6C	01101100	l	L
109	6D	01101101	m	M
110	6E	01101110	n	N
111	6F	01101111	o	O
112	70	01110000	p	P
113	71	01110001	q	Q
114	72	01110010	r	R
115	73	01110011	s	S
116	74	01110100	t	T
117	75	01110101	u	U
118	76	01110110	v	V
119	77	01110111	w	W
120	78	01111000	x	X
121	79	01111001	y	Y
122	7A	01111010	z	Z

ASCII decimal numbers 123-255 are 8-bit codes that represent special characters. To see these special characters and the key combinations that generate them, refer to the TRS-80 Model 100 Owner's Manual. See also Graph Key, Code Key, ASCII, CHR\$.

ASCII Code to Characters BASIC. See CHR\$.

ASCII Codes for Characters BASIC. See ASC.

ASCII Control Characters You can enter any possible ASCII character, 0-255, from the Model 100 keyboard using one or a combination of two or

ASCII Control Characters • Assembly Language

three keys. ASCII characters 33-255 display on the LCD, ASCII characters 0-32 do not. These are the control characters. Most of the control characters may be generated from the keyboard by pressing an ALPHA key and the CTRL key together. The remainder are generated by the BREAK/PAUSE and ESC keys, the spacebar, and the four cursor control keys.

Although you won't see the control characters you generate on the keyboard, you may get other results, such as the cursor jumping to another position or messages appearing onscreen, depending upon the application program you are using. This is because the program receiving the ASCII characters is watching for certain keys to be pressed to indicate various actions other than data entry. If you generate an ASCII control character with a code identical to that produced by a key which controls the program, the program will respond to the code rather than accepting it as data. For instance, you may use control codes to duplicate all of the codes generated by the special keys which control the TEXT program. See Control Codes, Text.

You may enter control characters to the computer in BASIC in the format:

PRINT CHR\$ (<n>)

<n> is the Control Character. See Control Characters, Computer.

The main purpose for control codes, however, is so that you can send commands or control codes to external devices such as a printer. By sending the right ASCII control code you can change print size, cause line feeds, etc. Normally, to do this you would enter a command line from BASIC in the format:

LPRINT CHR\$(<n>)

<n> is the ASCII decimal representation of a control character. See ASCII Control Characters—Printer.

ASCII Control Characters Characters having specific system-dependent meanings. Some of the control characters in the Model 100 are:

ASCII Code (decimal)	Meaning
7	bell
8	backspace (destructive)
9	tab
10	linefeed
11	home cursor
12	clear screen
13	carriage return
27	escape

To enter control characters to the computer, use the BASIC keywords, CHR\$ and PRINT. For example, entering:

PRINT CHR\$ (7)

causes the computer to BEEP.

ASCII Control Characters—Printer BASIC. To set the Epson RX-80 printer's print size, strike method, or number of lines-per-inch, the non-standard type format you want must be turned on by sending control codes to the printer. To do this, use an LPRINT statement containing ASCII codes and other special codes.

To return to the default ten characters-per-inch print size and six lines-per-inch, the non-standard type size currently in effect must be reversed. To turn non-standard print features on and off, use the ASCII codes in the BASIC command lines in the following table: (See table next page)

See Type Formats.

ASCII Keyboard Generally this keyboard includes three cases for each alpha character: upper case, lower case, and control. This keyboard provides keys for the set of 128 7-bit ASCII characters. The Model 100 keyboard will also generate the 8-bit ASCII characters, codes 123-255. You can do this using a combination of one standard qwerty keyboard key and the GRPH or CODE key or a combination of a standard key, GRPH or CODE key, and the SHIFT key all pressed at once. See Code Key, Graph Key.

ASR Automatic Send Receive. A terminal having not only a keyboard and printer, but also an automatic reading and recording device such as a cassette tape unit or a paper tape reader and punch.

Assembler A program which converts into binary object code the mnemonic form of the computer's language for execution. It acts as a compiler for Machine language and Assembly language programs.

Assembler Synonym for Assembly. See Assembly Language for a description of this programming language.

Assembly Language A programming language related very directly to the Machine language of the computer. Because the Model 100 uses an 80C85 processor chip, the Model 100 Assembler uses instructions based on 80C85 Machine language. The 80C85 handles information essentially one byte at a time. Larger units of data such as "character string," "floating-point numbers," and "records" are fictitious entities. In order to process data organ-

Assembly Language

ized in these structures, the 80C85 must execute many one byte instructions. For example, to move a 25 byte character string from one place to another, 25 one-byte moves must be performed. Of course, this would usually be done by a loop or using the Assembler's REP (repeat) function. Because of this, the Assembly language program is longer and more difficult to understand (and to code).

Originally only Assemblers were available for writing programs. The tedious and time-consuming nature of Assembly language programming led to the invention of "high-level" languages, such as BASIC, FORTRAN, COBOL, and Pascal. These languages greatly abbreviate the description of the

work to be done by the computer. A complex translation process is required to turn the high-level program into a Machine language program that the 80C85 can use. This translation process is time consuming and usually results in a Machine language program that runs many times slower than a corresponding program written in Assembly language. The compiled program is also usually many times larger than a corresponding program written in Assembly language. Because higher level languages are very general, the program procedures included can handle all possible circumstances in which the statements might be used. Since, for a particular program, only a small fraction of this generalized

ASCII Control Characters—Printer

Print Size

Print Type	Characters Per Inch	To Turn On (Enter BASIC Line)	To Turn Off (Enter BASIC Line)
Pica	10	LPRINT CHR\$(27) ; "P";	Default value
Elite	12	LPRINT CHR\$(27) ; "M";	LPRINT CHR\$(27) ; "P";
Enlarged	5	LPRINT CHR\$(27) ; CHR\$(14);	LPRINT CHR\$(20);
Condensed	17	LPRINT CHR\$(27) ; CHR\$(15);	LPRINT CHR\$(18);

Strike Method

Print Type	To Turn On (Enter BASIC Line)	To Turn Off (Enter BASIC Line)
Emphasized	LPRINT CHR\$(27) ; "E";	LPRINT CHR\$(27) ; "F";
Double Strike	LPRINT CHR\$(27) ; "G";	LPRINT CHR\$(27) ; "H";
Italics	LPRINT CHR\$(27) ; "4";	LPRINT CHR\$(27) ; "5";
Underline	LPRINT CHR\$(27) ; "-" CHR\$($\langle n \rangle$) ($\langle n \rangle$ is 1 or 49)	LPRINT CHR\$(27) ; "-" CHR\$($\langle n \rangle$) ($\langle n \rangle$ is 0 or 48)

Line Spacing

Line Spacing	To Turn On (Enter BASIC Line)	To Turn Off (Enter BASIC Line)
1/6 Inch	LPRINT CHR\$(27) ; "2";	Default Value
1/8 Inch	LPRINT CHR\$(27) ; "0";	LPRINT CHR\$(27) ; "2";
7/72 Inch	LPRINT CHR\$(27) ; "1";	LPRINT CHR\$(27) ; "2";
$\langle n \rangle$ /72 Inch	LPRINT CHR\$(27) ; "A"; CHR\$($\langle n \rangle$) $\langle n \rangle$ ranges from 0 to 85	LPRINT CHR\$(27) ; "2";
$\langle n \rangle$ /216 Inch	LPRINT CHR\$(27) ; "3"; CHR\$($\langle n \rangle$) $\langle n \rangle$ ranges from 0 to 255	LPRINT CHR\$(27) ; "2";

Assignment • Autodialing a Phone Number—TELCOM

code is needed, much of the code produced by the compiler code is not needed. It is generated because in some programs it would be needed. The information required to specify whether or not all of the possible circumstances need to be provided for the code is exactly what was left out to simplify the higher level language.

Interpreted languages (such as BASIC) have an additional performance disadvantage. They do the translation of higher level source code into executable Machine Code every time the program is executed, rather than only once.

Assembly language programs can run much faster, use less memory, and access more special machine level functions (like direct I/O devices) than BASIC or other high-level languages. Assembler is popular with software houses for writing programs that must run very fast (action games or programs to run fast devices such as disk drives, etc.). But because Assembler is harder to code, few individual users write programs in Assembler. Those who do ordinarily write only selected time-critical or highly specialized routines. More commonly, a program written largely in, for example, BASIC, may use a short section of code written in Assembler to do one particular time-critical or memory size critical task. The Assembler code will be called as a subroutine by the BASIC program.

In summary, Assembly language is a highly technical language more adapted to the needs of the 80C85 processor than to the needs of the Model 100 user. It is invaluable when speed or memory size is critical for highly specialized applications and for commercial software development. But for most Model 100 users it will not be practical to use Assembler as a regular programming language. They will rely on commercial software packages or write programs themselves in a high-level language such as BASIC. If an Assembly language program or subroutine is required for your application, paying an experienced Assembly language programmer would be considered an alternative to learning Assembler. But if you enjoy getting to the nuts and bolts of the computer, Assembly language is the quickest route to the inside workings of the 80C85. To learn more about Assembler, see *8080/Z80 Assembly Language Programming* by Alan R. Miller (John Wiley, 1981) and *Crash Course In Microcomputers* by Louis E. Frenzel, Jr. (Howard W. Sams & Co., Inc. 1980).

All 80C85 Assembly language mnemonics are defined in the encyclopedia for quick reference when you are writing programs in 8080/8085/80C85 Assembly language.

Assignment Giving a variable a value. In BASIC, a simple assignment would be:

X=5\$

This assigns the value 5 to the variable X.

Association for Computing Machinery See ACM.

Asynchronous An event or device which does not have the same timing as the central processing unit.

ATE Automatic Test Equipment. These devices and/or programs (usually ROM resident) automatically perform routine checks on equipment. The tests may occur in response to an event, such as powering on or resetting the system, or on a time scheduled basis.

ATN BASIC Function. ATN returns the arctangent of <number>. The format is:

<var> = ATN (<number>)

<number> is a numeric expression of any numeric type. ATN returns the angle whose tangent is <number>, the arctangent measured in radians in the range -pi to +pi where pi = 3.141593. To convert radians to degrees, multiply by 180/pi.

AT&T American Telephone and Telegraph.

Attenuation The reduction in strength of an electrical impulse.

Autodialing a Phone Number—TELCOM Once you have located and displayed a name and phone number on the LCD (using F1) in TELCOM entry mode, you have the choice of dialing it automatically or manually. To autodial, you need to have the Model 100 connected to a telephone using the direct connect modem cable. Then simply press F2, the call key in TELCOM entry mode. You will see the message "Calling" followed by the person's name. As the built-in modem pulse dials the number, the digits appear one by one on the same line following the person's name. If you are dialing so you can talk to the person on the other end, then pick up the receiver before all the digits are dialed. If you are calling another computer so it and the Model 100 can communicate, there is no need to pick up the receiver. You will need to press F4, the TERM or Terminal key in TELCOM Entry mode, to switch the computer over to Terminal mode, unless you have configured the phone number sequence in the address entry to do it for you automatically. Remember to set the ANS/ORIG switch on the left side panel of the computer to ORIG before entering terminal mode. See Modem Cable, TELCOM.

Autologon Using TELCOM • Autoplot*

Autologon Using TELCOM In many cases the host systems you access using TELCOM will require you to give your identification code and password before gaining access to the system. If you plan to use the modem cable and autodial the phone number for the host computer, automatically entering Terminal mode, you have the option of making the logon procedure automatic as well. To take advantage of this feature, you need to write an autolog sequence for the host system and include it in the appropriate address entry in your ADRS.DO file. Remember that this is a file created and updated in TEXT mode which contains one address per paragraph or record. Each record should begin with a name or code that you will use to locate the record, followed by the phone number bracketed with colons. To automatically enter Terminal mode when the number has been dialed, place the caret symbols (< >) following the phone number but before the second colon. The autolog sequence, if you choose to write one, goes between these carets (< >). The format is:

system name:phone number
<autolog sequence>:

The autolog sequence anticipates the routine opening prompts that the host system sends when you first access it and provides the appropriate responses. TELCOM has its own set of command characters, each one signifying an action to be taken by your computer during the autolog. The autolog sequence consists of a combination of these command characters and your ID number and password. Only the ID number and the password are sent to the host system; the rest of the sequence makes sure they are sent at the right time in the right way. The TELCOM key commands are:
?—The question mark is followed in an autolog sequence by an ASCII character. It tells the TELCOM program to wait for the host to send that character before proceeding with the rest of the logon.

=—The equal sign tells TELCOM to wait two seconds before proceeding with the autolog. It is good practice to include a pause after the phone number is dialed to give the phone connection a chance to become established before you send signals.

!—Any key command character in the autolog sequence that follows an exclamation mark will be interpreted as an ASCII character rather than a key command. For instance, if an autolog sequence contains != the equals sign will not cause a pause. It will simply be regarded as a normal character.

^—The caret key command makes the next character a control character. For instance, normally

pressing the CTRL key and M together is the same as pressing the ENTER key. In an autolog sequence, ^M has the same effect. It sends an ENTER signal to the host.

Let's say that after dialing the host computer and entering terminal mode, your manual logon goes like this. You initiate the logon by pressing the CTRL key and C together. The host system responds by prompting user id: at which point you key the user id 1235 and press ENTER. Then the host prompts Password: and you key and ENTER your password, rex. To make this process automatic you might write the following autolog sequence:

```
=^C?P1234^M?Urex^M$
```

This sequence starts with a two second pause (=), sends a control C(^C), waits for the letter U from the User id prompt (?U), then sends the user id# (1234) and an ENTER signal (^M). It then waits for the letter P from the Password prompt (?P), and sends the password (rex), followed by a second ENTER (^M). It is important to remember that TELCOM will wait for the exact character after a question mark found in an autolog sequence. For this reason, you must distinguish between capital and lower case letters. If the host sends you the prompt "user id" it will not work to wait for a lower case u. See Modem Cable.

Automatic Program Execution When you turn on the Model 100 the Main Menu Program automatically displays the menu screen and waits for an ENTER key press indicating which application program or RAM file you wish to use. If you position the cursor over the name of a RAM file containing an encoded BASIC program, indicated by the file extension .BA, and press ENTER, the BASIC PROGRAM will autorun. It is also possible to autorun a BASIC program rather than going to the Main Menu when the computer is first turned off, so that the chosen BASIC program will autorun the next time you power on.

First ENTER BASIC and WRITE or LOAD the program you want to autorun. Then SAVE the BASIC program to RAM using your choice of legal file names, six characters or less, beginning with a letter. Now key in the command:

IPL "filenm"

<filenm> is the name you gave to the BASIC file to autorun, and turn off the computer. The next time you power on, your BASIC program will automatically execute. See IPL.

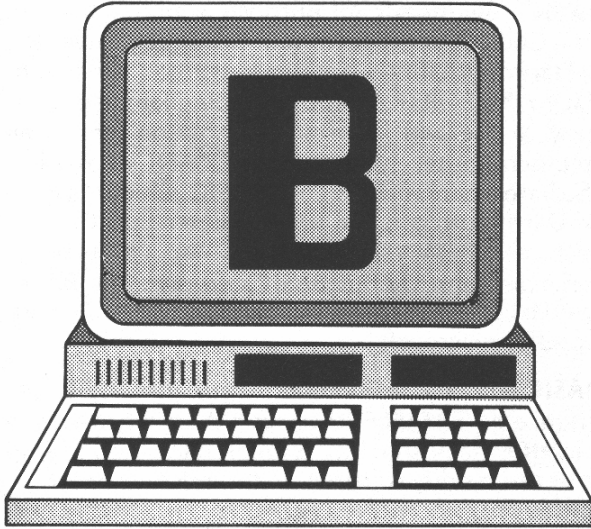
Autoplot* Color graphing software for the Model 100 (16K+) and the CGP-115 Color Graphic Printer, the Epson printer with Graphtrax, or the LNW-80. It

will plot functions or tabular data in a variety of graph formats, including bar and line graphs as well as scatter plots. The user may select a grid overlay, continuous curves and/or separate marks, plot size, numeric integration, and linear or log plots. The program scales and labels the axis automatically, and you may add legends and extra labels using an included label-maker program. Menly Systems.

Aux Use to hook up cassette tape. Before saving files on cassette the Model 100 must of course be connected with a cassette recorder. Radio Shack recommends that you use their CCR-81 Computer Recorder (RS-26-1208) which comes with its own connector cable (RS-26-1207). Before hooking up a cassette recorder, or any other peripheral device, it is good practice to turn the computer off.

The cassette cable itself consists of a large cylindrical plug on one end and three smaller plugs on the other, two of which are grey and the third black. The large cylindrical plug inserts into the leftmost socket, marked cassette, on the rear panel of the computer. Of the three smaller plugs on the other end of the cassette connector cable, the black one plugs into the EAR connector of the cassette recorder, the smaller of the two grey plugs inserts into the REM connector on the recorder, and the larger grey plug inserts into the cassette recorder's AUX connector. Having made these four connections, one to the computer and three to the tape recorder, you may turn the Model 100 back on, and then turn on the cassette recorder. You are now ready for cassette I/O. See Cassette I/O; Cassette TEXT File, Load; Cassette Save Instructions—TEXT File; CSAVEM; CLOADM; and CLOAD?

AV Available.



B ASCII 66, HEX 42. b—ASCII 98, HEX 62.

B B is used as an abbreviation for BYTES, when referring to memory; or for BAUD rate, when referring to communications. KB = 1000 bytes or baud (technically 1K = 1024 bytes).

B Bus line. Also, the hexadecimal symbol for the decimal number 11: B base 16 = 11 base 10 = 13 base 8 = 1011 base 2.

BA File Extension. When RAM files in the Model 100 are saved, they are automatically assigned one of three extensions by the operating system. When you save a file from BASIC it is assigned the extension, .BA, indicating that it is stored in tokenized, encoded form. Other extensions are .DO for an ASCII file, such as those created by and saved from TEXT, and .CO for Machine language files.

Background Program A low-priority program which is run when the processor has nothing else to do.

Backplane Also called a motherboard, it is the physical area in a system where the boards plug in. It usually contains the buses of the system in either printed circuit or wire-wrap form.

Backspace The cursor is the underline symbol (or with some programs a different symbol such as a square) which appears on the screen to let you know where an action (such as typing in a character, deleting, inserting, etc.) will take place. In the Model 100 the cursor is a blinking black or inverse video square.

In all modes the cursor moves to the right as you type and backs up to the left when you press the DEL/BKSP key, erasing the character to the immediate left of the cursor. To erase the character the

cursor is on, press the DEL/BKSP and SHIFT keys together.

Another way to backspace the cursor is to press the ← key. In most modes this will erase the character the cursor backspaces over. In TEXT mode, and TEXT, as invoked by the BASIC EDIT command, however, backspacing the cursor with the ← key does not erase the characters the cursor passes over. Rather it provides a means of positioning the cursor within the text to indicate where other actions such as inserting text or marking a text block will take place.

Backspace (←) Pressing the left arrow key moves the cursor left one character: holding down the ← key causes this process to autorepeat, scrolling left along the current line. In some applications, when the cursor reaches the leftmost position of the current line it wraps around to the rightmost position of the previous line to repeat the process. As long as the ← key is held down, the cursor continues to move to the left and in the direction of the beginning of the file. In other cases, cursor left movement is restricted to the current line. In all modes except TEXT and BASIC EDIT, backspacing the cursor with the ← key erases the characters the cursor passes over.

In TEXT, and in TEXT as invoked by the BASIC EDIT command, the ← key performs additional functions if used in conjunction with the SHIFT and CTRL keys. Pressing Shift and ← moves the cursor to the first character of the current word. If it is already there, the cursor will move to the first character of the next word to the left. Pressing CTRL and ← moves the cursor to the first character of the current line.

Backspace One Word In TEXT mode, and TEXT as invoked by the BASIC EDIT command, you may backspace the cursor one word by pressing the ← and SHIFT keys together.

Backup Copy A duplicate copy of a program or data stored on a different diskette or cassette, in case of loss or damage to the original.

Backup Diskettes Because diskettes can be rendered unreadable by physical damage, magnetic contamination, or dirt, it is wise to keep at least two copies of any important information. See Write-Protect, Adhesive Tab.

Bad Sector A sector on the diskette which will not read/write data correctly. Usually due to a minor physical flaw in the diskette. One or two bad sectors are not serious—DOS marks them as bad and avoids using them. More than a few bad sectors

Bank • BASIC—Statements Cross-Referenced by Function:

indicates the diskette should be used as a frisbee rather than to hold valuable data.

Bank Usually 64K of memory.

Bank Select Also called bank switching. A method of extending a computer's RAM memory. Each bank responds to the same addresses, but only one is active at a time.

Bar Code The consumer product information code which is put on products read by an optical wand. The Model 100 has a bar code reader interface located on the left side panel of the computer, closest to the rear. It uses a Hewlett-Packard HEDS-3000 compatible bar code reader. However, there is currently no software available to support this feature.

Bar Code Reader* The Alpha 100 Bar Code reader is compatible with the Model 100. Software not included. Alpha 100.

Base Register The register which contains the longer part of a two part Assembly language address. The smaller part, called the displacement or offset, is contained in the Assembly language instruction. The data that the instruction is to operate on is located at the "effective address"—the byte of memory whose address is the sum of the number in the specified base register and the offset number given in the instruction. This method of specifying addresses allows data to be located with fewer bits than with a full, explicit address. The model 100's 80C85 processor does not use a base register.

BASF Badische Anilin und Soda Fabrik, a manufacturer of magnetic storage media including diskettes.

BASIC Beginner's All-purpose Symbolic Instruction Code. A popular computer language invented at Dartmouth for educational purposes. While similar to the FORTRAN programming language, it is easy to use and learn and is now on almost all microcomputer systems. Some BASICs have just the bare essentials of regular BASIC, which is a form of Dartmouth original BASIC. Super BASICs have features other versions do not. Compatibility problems between various BASICs do exist. The BASIC used by the Model 100 is an extended BASIC developed by Microsoft.

BASIC—LOAD Instructions You can LOAD a program, e.g. "SAMPLE" that was SAVED in RAM or on cassette, back into the Model 100's memory, in order to modify or run it. If it is in RAM, enter:

BASIC

then enter

LOAD "SAMPLE"

You may use F2, the Load key, to automatically type the "LOAD" portion of the command.

If SAMPLE is on cassette, enter BASIC, rewind the cassette, press PLAY and key:

LOAD "CAS:SAMPLE"

or

CLOAD "SAMPLE"

All LOAD instructions erase any program lines you have in memory before the LOAD. To combine a SAVED program with the one you are writing, use MERGE (q.v.).

To automatically run the program after it is LOADED, use LOAD "SAMPLE",R; or RUN "SAMPLE",R. See Automatic Program Execution, LOAD, RUN.

BASIC—Statements Cross-referenced by Function:

Absolute value of number	ABS(<numex>)
Address of variable in memory	VARPTR(<variable>)
Arctangent	ATN(<numer>)
Arrays	DIM <variable><dimension>[<dimension>,...]>[,<variable>(<dimension>[,<dimension>,...]>)]
ASCII codes for characters	ASC(<string ex>)
ASCII code to characters	CHR\$(<numeric>)
Branch	GOTO<line num>
Call BASIC subroutine	GOSUB <linenum> and RETURN
Carriage position, printer	LPOS(<dumy number>)
Cassette—save a BASIC program	CSAVE "<filename>" [,A]

BASIC Statements—Cross-Referenced by Function:

BASIC—Statements Cross-referenced by Function:

Cassette, load BASIC program to	CLOAD ["<filename>" [,R]]
Cassette file, verify load of	CLOAD? ["<filename>"]
Cassette motor control, turn on/off automatically	MOTOR<on or off>
Change a BASIC line	EDIT<line num>
Change memory byte with BASIC	POKE<memory address,byte val>
Character—read from keyboard	INKEY\$
Clear the screen	CLS
Close open data files	CLOSE [<filenum list>]
Communications interrupt(modem)—enable/disable	MDM <ON,OFF, or STOP>
Communications interrupt(RS-232C)—enable/disable	COM <ON,OFF, or STOP>
Communications interrupt(RS-232C)—define	ON COM GOSUB <linenum>
Communications interrupt(modem)—define	ON MDM GOSUB <linenum>
Convert decimal number to integer (by truncation)	CINT(<numex>)
Cosine of an angle	COS(<numex>)
CPU port input data from	INP(<port num>)
CPU output byte to port	OUT <portnum>,<byteval>
Cursor, find current location of	CSRLIN and POS(<dummy numex>)
Date, set system value	DATE\$ = "<month>,<day>,<year>"
Day of the week—set system value	DAY\$ = "<day var>"
Double-precision number, convert to	CDBL(<numex>)
e, powers of	EXP(<numex>)
Erase a RAM file	KILL"<filename>"
Erase current BASIC program and variables	NEW
Erase current BASIC variables, (not program) allocate string space, and protect high memory	CLEARL,[<string space>,<high mem>]
Error—line number where it occurred	ERL
Error code number, last	ERR
Execute BASIC program	RUN
Execute Machine language program	RUNM
Exit BASIC	MENU
Exponential	EXP(<numex>)
Files—define number in BASIC	MAXFILES = <numex>
Function key interrupt, define	ON KEY GOSUB <linenum list>
Function keys, define value of	KEY <key num>,<string ex>

BASIC Statements—Cross-Referenced by Function:

BASIC Statements—Cross-Referenced by Function:

Function keys, enable/disable interrupt	ON(<keynum>) <ON,OFF, or STOP>
Function keys, list current values of	KEY list
Graphics—draw line, box	LINE (<xcoord1>,<ycoord1> <xcoord2>,<ycoord2>)[,<switch>][,B[F]]
Graphics, pixel	PRESET(<xcoord>,<ycoord>) and PSET(<xcoord>,<ycoord>)
Immediate response to one-character answers (without using enter key)	Y or N, drive letter, etc. INKEY\$
Integer—convert to, by rounding	CINT(<numex>)
Interrupt clock, define	ON <hour,min,sec> GOSUB <linenum>
Interrupt error handling, define	ON ERROR GOTO <linenum>
Interrupt function key, define	ON KEY GOSUB <linenum list>
Interrupt modem, define	ON MDM GOSUB <linenum>
Interrupt modem, enable/disable	MDM<ON, OFF, or STOP>
Interrupt RS-232-C, enable/disable	COM <ON, OFF, or STOP>
Interrupt RS-232-C, define	ON COM GOSUB <linenum>
Keyboard—input line of string data	LINE INPUT [“<prompt>”];<string var>
Keyboard, read characters from	INKEY\$
Keyboard, read data from	INPUT[“<prompt>”];<variable>[,<variable>,...]and INPUT\$(<numex>)
Line number where error occurred	ERL
List program on LCD	LIST [line1][-line2]
Load a BASIC program from a device	LOAD [“<device:>”][“<filename or configuration>”][,R]
Load binary data (Machine language programs, etc.) from cassette	CLOADM[“<filename>”]
Location of cursor, find	CSRLIN and POS(<dummy numex>)
Machine language program, load	LOADM “[<device:>]<filename>”
Machine language program, save	SAVEM “<device:> <filename>,” <start address>,<end address>[,<entry address>]
Machine language program (subroutine)	CALL <address>[<accumlatervel>,<HL val>]
Machine language program, save on cassette	CSAVEM “<filenam>”,<start address>,<end address>[,<entry address>]
Memory address of variable	VARPTR(<variable>)
Memory, amount free	FRE(<dummy numex>)
Memory, change	POKE<memory address> <byte val>
Memory, read byte of	PEEK(<memory address>)
Music	SOUND <frequency>,<duration>
Natural logarithm	LOG(<numex>)

BASIC Statements—Cross-Referenced by Function:

BASIC Statements—Cross-Referenced by Function:

Next, end of FOR...NEXT loop	FOR <cartvar> = <start val> TO <end val> [STEP<increment>]
Number, convert from string to	VAL(<string ex>)
Position of printer carriage	LPOS(<dummy numex>)
Print BASIC program listing on printer	LLIST[<line1>][-<line2>]]
Print on printer	LPRINT<list of expressions>(separate items with commas or semicolons)
Print screen contents on printer	LCOPY
Printer, carriage position of	LPOS(<dummy numex>)
Program, BASIC—run or execute	RUN
RAM file, rename	NAME “[RAM:]<oldname>” AS “[RAM:] <newname>”
RAM files, list	FILES
Random number	RND(<numex>)
Read character from keyboard	INKEY\$,INPUT[“<prompt>”:]<varlist>(separate items with commas)
Read data from file	INPUT# <file num>,<var list>(separate items with commas)
Read data from program line	READ <vat list> (separate items with commas, use with DATA statements.)
Rename a RAM file	NAME “[RAM:]<oldname>” AS “[RAM:] <newname>”
Repeat a character (n) times	STRING\$(<n>,<char>)
Repeat program lines	FOR <carntvar> = <start val> TO <end val> [STEP<increment>]
Resume program execution after a STOP or BREAK	CONT
Save a BASIC program	SAVE “[<device:>]<filename or configuration>”[,A]
Save binary data	SAVEM“<device:> <filename>,<start address>,<end address>[,<entry address>]
Screen, clear	CLS
Screen dump to printer	LCOPY
Search string for character or shorter string	INSTR(<start position>,<search string>, <match string>)
Sign	SGN(<numex>)
Sine of an angle	SIN (<numex>)
Single-precision numbers, convert to	CSNG(<numex>)
Speaker	BEEP AND SOUND <frequency>,<duration>
Square Root	SQR(<numex>)

BASIC, Largest Line Number in • BASIC, Modes of Operation in

BASIC Statements—Cross-Referenced by Function:

Stop BASIC program	END
String, convert to number	VAL(<string ex>)
String, length of	LEN(<string ex>)
String, numeric value of	VAL(<string ex>)
String, return left portion	LEFT\$(<string ex>,<numex>)
String, return or replace characters	MID\$(<string ex>,<start position>[,<length>])
Subroutine	GOSUB <linenum>
Subroutine, Machine language	CALL <address>[,<numex>,<numex>]
Tangent of an angle	TAN(<numex>)
Time, set system value	TIME\$ = "<hour>:<minute>:<second>"
Variable, define as double-precision	DEFDBL: <letter list> (separate items with commas.)
Variable, define as integer	DEFINT <letter list> (Separate items with commas.)
Variable, define as single precision	DEFSNG: <letter list> (separate items with commas.)
Variable, define as string	DEFSTR <letter list> (Separate items with commas.)
Verify cassette file load	CLOAD? ["<filename>"]

BASIC, Largest Line Number in The largest possible line number for a BASIC program is 65529.

BASIC, Modes of Operation in The BASIC that comes with the Model 100 is an Extended BASIC created by Microsoft. BASIC has three modes of operation: Command mode, Execute mode, and Edit mode. When you first enter BASIC from the Main Menu you will be in Command mode.

Command mode displays the prompt OK to indicate that you may enter BASIC commands or BASIC program lines. Program lines always begin with a program line number. Program lines may be numbered between 0 and 65529. When you key in a program line and ENTER it to BASIC it is stored in BASIC memory and not executed or acted upon by the computer, until you enter the BASIC RUN command (or its equivalent). When you RUN or execute the program lines, they are executed in numerical order from first to last line, unless you have included BASIC statements to branch execution to nonsequential program lines.

To list the program lines currently stored in BASIC memory on the LCD, key and ENTER the BASIC command LIST. To list program lines on an attached printer, key and enter LLIST.

BASIC commands may be entered without program lines while in Command mode. In this case, when you enter the command line to BASIC, it is immediately executed. The computer immediately

performs the actions represented by the keywords you have entered. Commands entered without line numbers are not saved to BASIC memory and will not LIST.

Execute mode is the state the computer is in when you run a BASIC program or enter a command line. The computer is interpreting and acting upon the BASIC keywords in the program or command line. While BASIC is executing, you may not enter characters from the keyboard unless BASIC requests them. The BASIC prompt to enter data from the keyboard is "?". If the programmer has done a good job, the program will also print a prompt to tell the user what kind of data to input.

To temporarily pause program execution, press the BREAK/PAUSE key once. To resume execution at the program line where it was paused press the BREAK/PAUSE key a second time. The computer will not respond to any key pressed while program execution is paused in this way.

To break program execution, press the SHIFT and BREAK/PAUSE keys together. This displays the message Break in <xxxx>, where <xxxx> is the program line that was executing at the time of the break. You will now be in command mode and may use command lines (such as print and assignment statements) to display and change variable values. To resume program execution with the line that was executing at the time of the break, enter the BASIC command CONT. Program execution will

not resume if you have altered any program lines during the break.

EDIT mode must be entered to alter any program line that has already been ENTERed into BASIC memory. Program lines that have not yet been ENTERed may be edited by backspacing and rekeying the line or line segment. To edit any other line, key and ENTER the BASIC command EDIT. This translates the contents of BASIC memory, which are stored in a tokenized, compressed, format into a temporary ASCII format file in RAM. You will see the first eight lines of the program on the LCD screen and TEXT mode will be enabled, allowing you to use all its editing functions on the program.

When you are finished using the text processor to edit the BASIC program, press F8 (Exit key) in BASIC EDIT mode. This translates the program lines in your temporary edit file from ASCII format back to tokenized BASIC form in BASIC memory. You will see the OK prompt indicating you are back in BASIC Command mode. Now if you LIST the program you will see the changes you made while editing.

If you have previously SAVED a copy of the BASIC program to a RAM file, the editing changes you made will also be reflected there. There is no need to SAVE the newly edited copy of the program back to the BASIC format RAM file. If you want to keep an unchanged copy of the program as well as an edited version you should SAVE the BASIC program to a RAM file in ASCII format before editing the copy in BASIC memory. The editing changes you make to the BASIC program will not be reflected in the ASCII version in RAM memory. This copy may later be LOADED to BASIC to restore it to the tokenized format BASIC uses.

BASIC, NEC PC-8201A The BASIC included was written by Microsoft and is almost identical to the MODEL 100 BASIC. The PC-8201A version offers the LOCATE and RENUM commands which are not offered by the Model 100 version. PC-8201A BASIC also makes all numeric variables single-precision unless specified otherwise by DEFINT or DEFDBL statements.

BASIC, NEW Programs in To erase old program lines and start a new program, enter:

NEW

This completely erases all lines currently in BASIC's memory, so if it's something you want to keep, and you haven't already got a copy elsewhere, SAVE it first. If you don't erase the program in memory before starting on another, you will usually wind up

with an unusable combination of mixed lines from your old and new programs.

BASIC, Prompts in OK and ? are prompts from the BASIC language. OK is the prompt from BASIC indicating that you can now enter a BASIC command or statement. ? is the prompt from a program written in the BASIC language which is running and needs you to type in data to answer a question. In this case it is a good idea to include a description such as "Enter check amount" in the program so the operator (or you) will know exactly what should be entered in response to the ? prompt. An example of how to do this:

```
100 INPUT "ENTER CHECK AMOUNT"  
    ,CHECKAMOUNT
```

this will give the operator using the program this prompt upon the screen:

```
ENTER CHECK AMOUNT ?
```

BASIC, SAVEing a Program in To save a program named "SAMPLE" to RAM, enter:

```
SAVE "SAMPLE"
```

You may use F3 to type the SAVE portion for you. On cassette, first rewind the cassette, press PLAY and RECORD together, then enter:

```
SAVE "CAS:SAMPLE"
```

To run the program at a later time, use the LOAD command to copy it from the cassette or RAM file you SAVED it to, back into BASIC. See LOAD. If you want to SAVE the program in ASCII format, which allows you to MERGE two programs together, put ,A after the close quote of the program name. For example:

```
SAVE "SAMPLE",A
```

See MERGE.

The SAVE instruction does not alter your program in memory. It is important to be aware that if you write a BASIC program, it will be lost (erased) unless you SAVE it before you use the NEW command or LOAD a new program to BASIC.

BASIC, Special Characters in The following characters have special meanings in BASIC. These symbols may not be used as part of variable names, and many of the symbols have special functions or other uses which vary from BASIC statement to BASIC statement.

	blank space
=	equal sign or assignment symbol
+	plus sign or concatenation symbol
-	minus sign

BASIC Command Table • BASIC EDIT

/	slash or division symbol
\	backslash or integer division symbol
^	caret or exponential symbol
*	asterisk or multiplication symbol
%	percent sign or integer type declaration character
#	number (or pound) sign, or double precision type declaration character
\$	dollar sign or string type declaration character
!	exclamation point single-precision type declaration character
&	ampersand
,	comma
'	single quotation mark delimiter (apostrophe) or remark point
.	period or decimal delimiter
;	semicolon
:	colon or statement separator
?	question mark or PRINT abbreviation
_	underline
"	double quotation mark or string delimiter
<	less than
>	greater than
(left parenthesis
)	right parenthesis

BASIC Command Table

Command	Operation
CLEAR	Erases all variable values, protects high memory
CLS	Clears the screen, returns cursor to upper left of LCD
CONT	Resumes program execution after a break
FILES	Lists RAM file names on LCD
FRE	Returns current amount of BASIC memory
HIMEM	Returns highest memory address available to BASIC
IPL	Defines a BASIC program to execute first when you power on
KILL	Deletes a RAM file
LCOPY	Prints the LCD screen contents on the printer

Command	Operation
LIST	Lists the current contents of BASIC memory on the LCD
LLIST	Lists the contents of BASIC memory on the printer
LOAD	Loads a BASIC program from a device file
MAXRAM	Returns highest RAM memory address
MENU	Exits BASIC & returns to the Main Menu
MERGE	Merges program lines in BASIC memory with an ASCII program file
NAME	Renames a RAM file
NEW	Deletes the current contents of BASIC memory
POWER	Sets, executes, and disables the automatic power off feature
?	Shorthand for the BASIC PRINT command
SAVE	Saves the current contents of BASIC memory to a device file

BASIC Compiler A program which accepts BASIC programs as input and translates them into the actual numeric instruction code of the 80C85 microcomputer chip. These translated versions of programs are called object programs because they are in object code.

BASIC EDIT EDIT mode must be entered to alter any program line that has already been ENTERed into BASIC memory. Program lines that have not yet been ENTERed may be edited by backspacing and rekeying the line or line segment. To edit any other line, key and ENTER the BASIC command EDIT. This translates the contents of BASIC memory, which are stored in a tokenized, compressed, format into a temporary ASCII format file in RAM. You will see the first eight lines of the program on the LCD screen and TEXT mode will be enabled, allowing you to use all its editing functions on the program.

When you are finished using the text processor to edit the BASIC program, press F8 (EXIT key) in BASIC EDIT mode. This translates the program lines in your temporary edit file from ASCII format back to tokenized BASIC form in BASIC memory. You will see the Ok prompt indicating you are back in BASIC command mode. Now if you LIST the pro-

BASIC Error Code Numbers • BASIC File I/O Commands and Functions

gram you will see the changes you made while editing.

If you have previously SAVED a copy of the BASIC program to a RAM file, the editing changes you made will also be reflected there. There is no need to SAVE the newly edited copy of the program back to the BASIC format RAM file. If you want to keep an unchanged copy of the program as well as an edited version you should SAVE the BASIC program to a RAM file in ASCII format before editing the copy in BASIC memory, using the EDIT command. The editing changes you make to the BASIC program will not be reflected in the ASCII version in RAM memory. This copy may later be LOADED to BASIC to restore it to the tokenized format BASIC uses.

BASIC Error Code Numbers

Error Number	Error Message	Error Condition
1	?NF Error	NEXT without FOR
2	?SN Error	Syntax Error
3	?RG Error	RETURN without GOSUB
4	?OD Error	Out of Data
5	?FC Error	Illegal Function Call
6	?OV Error	Overflow
7	?OM Error	Out of Memory
8	?UL Error	Undefined Line
9	?BS Error	Bad Subject
10	?DD Error	Double Dimensioned Array
11	?/0 Error	Division by Zero
12	?ID Error	Illegal Direct
13	?TM Error	Type Mismatch
14	?OS Error	Out of String Space
15	?LS Error	String Too Long
16	?ST Error	String Formula Too Complex
17	?CN Error	Can't Continue
18	?IO Error	Error
19	?NR Error	No RESUME
20	?RW Error	RESUME Without Error
21	?UE Error	Undefined Error
22	?MO Error	Missing Operand
23-49	?UE Error	Undefined Error
50	?IE Error	Undefined Error
51	?BN Error	Bad File Number
52	?FF Error	File Not Found
53	?AO Error	File Already Open
54	?EF Error	Input Past End of File
55	?NM Error	Bad File Name
56	?DS Error	Direct Statement in File
57	?FL Error	Undefined Error
58	?CF Error	File Not Open
59-255	?UE Error	Undefined Error

BASIC Error Messages

Error Message	Error Number	Error Condition
?AO Error	53	File Already Open
?BN Error	51	Bad File Number
?BS Error	9	Bad Subscript
?CF Error	58	Closed File Accessed
?CN Error	17	Can't Continue
?DD Error	10	Doubly Dimensioned Array
?DS Error	56	Direct Statement in File
?EF Error	54	Input Past End of File
?FC Error	5	Function Call Illegal
?FF Error	52	File Not Found
?FL Error	57	Undefined
?ID Error	12	Illegal Direct
?IE Error	50	Undefined
?IO Error	18	Error
?LS Error	15	String Too Long
?MO Error	22	Missing Operand
?NF Error	1	NEXT without FOR
?NR Error	19	No Resume
?NM Error	55	File Name Bad
?OD Error	4	Out of Data
?OM Error	7	Out of Memory
?OS Error	14	Out of String Space
?OV Error	6	Overflow
?RG Error	3	RETURN without GOSUB
?RW Error	20	Resume without Error
?SN Error	2	Syntax Error
?ST Error	16	String Too Complex
?TM Error	13	Type Mismatch
?UE Error	21,23-49, 59-255	Undefined Error
?UL Error	8	Undefined Line

BASIC File I/O Commands and Functions

Device Names						
BASIC Command or Function	RAM:	CAS:	COM:	MDM:	LCD:	LPT:
CLOAD	@	I				
CLOAD?	@	[]				
CLOADM	@	I				
CLOSE	#	I/O	#	I/O	#	O # O
CSAVE	@	O				
CSAVEM	@	O				
INPUT #	#	I	#	I	#	I
KILL	@	[]				

(table continues on next page)

BASIC Files, Open • BASIC Input/Output Commands by Device

Device Names						
BASIC Command or Function	RAM:	CAS:	COM:	MDM:	LCD:	LPT:
LINEINPUT#	# I	# I	# I	# I		
LOAD	@ I	I	I	I		
LOADM	@ I	I				
MERGE	@ I	I	I	I		
NAME	@ []					
OPEN	# I/O	# I/O	# I/O	# I/O	# O	# O

KEY

I = input to the computer only

O = output from the computer only

I/O = input/output

= references the device by using an associated file number, device, file number, and file name or

BASIC Files, Open The BASIC RUN command has optional specification, R. If you append an R separated by a comma from the first segment of the RUN command, the currently open files remain open. RUN without the ,R option closes all currently open files before RUNNING a new program. It is useful to keep the files used by the previous program open for the new one in cases where the two programs use common files.

Device Names						
BASIC Command or Function	RAM:	CAS:	COM:	MDM:	LCD:	LPT:
PRINT#	# O	# O	# O	# O	# O	# O
PRINT#USING	# O	# O	# O	# O	# O	# O
RUN	@ I	I	I	I		
RUNM	@ I	I				
SAVE	@ O	O	O	O	O	O
SAVEM	@ O	O				

configuration are associated by the OPEN statement and terminated by the CLOSE statement.

@ = default device, redundant to specify this device with this statement or command

[] = not I/O but file related (verify, delete, rename)

The formats are:

RUN ,R

or

RUN <line> ,R

<line> is the program line number where execution is to begin. If <line> is not specified, execution begins with the lowest program line number.

BASIC Input/Output Commands by Device

See table below.

Cassette	
I/O Commands	Function Performed
CLOAD	Loads a tokenized BASIC format program from cassette
CLOAD?	Compares a cassette copy of a file with the copy loaded to BASIC
CLOADM	Loads a Machine language format file from cassette
CSAVE	Saves the current contents of BASIC memory to cassette
CSAVEM	Saves a Machine language program from internal memory to cassette tape
MOTOR	Turns on and off the cassette tape motor
CPU	
I/O Commands	Function Performed
OUT	Outputs a data byte to the central processing unit
INP	Inputs a data byte from the central processing unit
INPUT	Prompts for data input from the keyboard
INPUT\$	Assigns a string of a given length, input from the keyboard, to a string variable
INKEY\$	Accepts the string value of the key currently pressed as variable data
LINE INPUT	Assigns a line of data, input from the keyboard, to the a string variable

BASIC Input/Output Commands by Device

BASIC Input/Output Commands by Device continued

ON KEY GOSUB	Defines an interrupt subroutine to execute when a specific function key is pressed
LCD	
I/O Commands	Function Performed
CLS	Turns off all LCD screen pixels
CSRLIN	Gives cursor's column position on the LCD
LCOPY	Prints current contents of the LCD to the printer
LINE	Draws a line or box on the LCD between coordinate points
POS	Gives cursor's line position on the LCD
PRESET	Turns off the LCD pixel at the screen coordinates given
PRINT	Prints data beginning at the current cursor position
PRINT@	Prints data at a specified LCD screen location
PRINT USING	Prints data on the LCD using a specified format
PSET	Turns on the LCD pixel at the screen coordinates given
SCREEN	Turns the function key label line (8) on and off
TAB	Tabs the cursor to the given LCD screen position
Printer	
I/O Commands	Function Performed
LCOPY	Prints current screen contents on the printer
LLIST	Prints all or a part of the current contents of BASIC memory on the printer
LPRINT	Prints data on the printer
LPRINT USING	Prints data on the printer using a specified format
LPOS	Returns the current position of the printer head
TAB	Prints the next data beginning at the line position specified
Sound Generator	
Output Commands	Function Performed
BEEP	Makes the computer beep
SOUND	Makes the computer emit a tone of given pitch and duration
SOUND ON	Makes the computer emit a tone when loading a file from cassette and when waiting for the carrier signal in telecommunications
SOUND OFF	Disables SOUND ON function

BASIC Interrupt Commands

See next page for table.

BASIC Input/Output Commands by Device

BASIC Interrupt Commands

Interrupt Command	Enable	Disable	Hold	Interrupt Operation When Enabled
ON COM GOSUB<line>	COM ON	COM OFF	COM STOP	Branches execution to a subroutine when RS-232-C activity occurs
ON ERROR GOTO<line>				Branches to an error handling routine in an error situation
ON KEY GOSUB<line>	KEY ON	KEY OFF	KEY STOP	Branches to subroutine defined for a function key when that key is pressed
ON MDM GOSUB<line>	MDM ON	MDM OFF	MDM STOP	Branches to a subroutine when modem activity occurs
ON TIME\$ =<time> GOSUB	TIME\$ON	TIME\$OFF	TIME	Branches to a subroutine when the <time> equals the system time

BASIC Logical Generator Table

Logical	Description	First (fbit)	Second (sbit)	Result (rbit)
AND	If fbit and sbit are both 1, rbit is 1	1	1	1
		0	1	0
		1	0	0
		0	0	0
OR	If either fbit or sbit is 1, rbit is 1	1	1	1
		0	1	1
		1	0	1
		0	0	0
XOR	If either bit is 1 while the other bit is 0, the rbit is 1	1	1	0
		0	1	1
		1	0	1
		0	0	0
EQV	If both bits have the same value, rbit is 1	1	1	1
		0	1	0
		1	0	0
		0	0	1
IMP	If fbit is 1 and sbit is 0, rbit is 0, otherwise rbit is always 1	1	1	1
		0	1	1
		1	0	0
		0	0	1
NOT	If fbit 0 rbit is 1	1		0
		0		1

BASIC Math Operators Table • BASIC Program Lines, Display

BASIC Math Functions Table

Math Function	Operation Performed
ABS	Absolute value of a number is returned
ASC	ASCII code is found
ATN	Arctangent calculation is performed
CDBL	A number is converted to double-precision
CINT	Conversion to an integer
COS	Cosine calculation is performed
CRSLIN	Gives vertical, line number, position of the cursor
CSNG	A number is converted to single-precision
EOF	End-of-file status is returned
ERL	Shows the line number where the last error occurred
ERR	The error code of the last error is shown
EXP	Exponential calculation
FIX	Convert to integer by truncation
FRE	Gives the number of unused bytes of RAM memory
HIMEM	Returns highest memory address available to BASIC
INP	Reads a value from the CPU port
INSTR	Searching for a substring from a string
INT	Integer conversion
LEN	Find string length
LOG	Natural logarithm is calculated
LPOS	Gives carriage position of the printer
MAXRAM	Gives the highest RAM memory address, size
PEEK	Memory address value is shown
POS	Gives the current cursor column position
RND	Pseudo-random number is given
SGN	Gives sign, in algebraic form
SIN	Sine is calculated

Math Function	Operation Performed
SQR	Square root is calculated
TAB	Cursor or print head positioning
TAN	Tangent is calculated
VAL	Returns the numeric value of the string
VARPTR	Memory address of a variable is given

BASIC Math Operators Table

Math Operator	Operation Performed
+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Integer Division
^	Exponentiation
MOD	Modulus Arithmetic

BASIC On One Line You can put a colon (:) at the end of one statement and continue entering another statement on the same line, without giving a new line number. The line number at left refers to all statements on the line. An apostrophe (') allows a comment or remark to be added after a statement with or without a separating colon. Examples:

```
40 LET X = 1:REM Comment
50 LET X = 1:LET Y = 2
60 LET X = 1' Comment
```

BASIC Program Lines, Delete To delete a BASIC program line that has not yet been ENTERed, backspace over the line using the DEL/BKSP or ← key. Or, pressing the SHIFT and BREAK/PAUSE key combination will also cancel the current line. If the program line has been ENTERed to BASIC, simply type and ENTER the line number you want to erase. This replaces the line with a blank line which is essentially ignored when you run the program.

BASIC Program Lines, Display To display all program lines, enter:

LIST

To display program lines from start up to line 100, enter:

LIST -100

To display program lines from line 100 to end, enter:

LIST 100-

BASIC Reserved Words, • BASIC Statements—Table of Formats, Descriptions

To display program lines from line 100 to line 200, enter:

LIST 100-200

To display program line 100 only, enter:

LIST 100

To freeze the list while it is being displayed so you can read it, press the BREAK/PAUSE key once. To restart listing, press BREAK/PAUSE a second time.

BASIC Relational Operators Table

Relational Operator	Function
<	Less than
>	Greater than
=	Equal to
≠ or ><	Not equal to
=> or ≥	Greater than or equal to
=< or ≤	Less than or equal to

BASIC Reserved Words, Uses and Restrictions of
Reserved words have particular meanings in BASIC and are used for commands, statements, functions and operator names. These words cannot be used as variable names, and variable names cannot be used as reserved words followed by any of the type declaration characters—\$, %, !, #. Even embedding a reserved word, such as RUN, embedded in RUNT, a variable name, could cause problems.

The reserved words, when used, are to be delimited (separated with space or spaces around them) so they are easily recognized by BASIC. If you accidentally use one of the reserved words, you may see error message or have strange results when you execute the program. See also Debug, or Test a Program.

Following are the reserved words in BASIC:

ABS	AND	ASC
ATN	BEEP	CALL
CDBL	CHR\$	CINT
CLEAR	CLOAD	CLOADM

CLOSE	CLS	COM
CONT	COS	CSAVE
CSNG	CSRLIN	DATA
DATE\$	DAY\$	DEF
DIM	DSK\$	DSKO\$
EDIT	ELSE	END
EOF	EQV	ERL
ERR	ERROR	EXP
FIX	FILES	FOR
FRE	GOTO	GOSUB
HIMEM	IF	IMP
INKEY\$	INP	INPUT
INSTR	INT	IPL
KEY	KILL	LCOPY
LEFT\$	LEN	LET
LFILES	LINE	LIST
LLIST	LOC	LOF
LOAD	LOG	LPOS
LPRINT	MAX	MERGE
MDM	MENU	MID\$
MOD	MOTOR	NEXT
NAME	NEW	OFF
NOT	ON	OPEN
OR	OUT	PEEK
POKE	POS	POWER
PSET	PRESET	PRINT
READ	REM	RESTORE
RESUME	RETURN	RIGHT\$
RND	RUN	SAVE
SCREEN	SGN	SIN
SOUND	SPACES\$	STEP
SQR	STOP	STR\$
STRING\$	TAB	TAN
THEN	TIMES\$	TO
USING	VAL	VARPTR
WIDTH	XOR	

BASIC Statements—Table of Formats, Descriptions:

Type:

C = Command
S = Statement (not Input/Output)
S-I/O = Input/Output Statement
F = Function

Statement	Type	Description
ABS(<numvar>)	F	Absolute value of <numvar>
ASC(<string ex>)	F	Gives ASCII code of 1st character of <string ex>
ATN(<numex>)	F	Returns the arctangent of <numex>
BEEP	S-I/O	Speaker emits beep sound
CALL<address>, [<exp1>],[<exp2>]	S	Machine language program called, up to two arguments passed

BASIC Statements—Table of Formats, Descriptions

Statement	Type	Description
DEFSNG<letterlst>	S	Defines all variables beginning with the letters in the <letterlst> as integer numeric variables
DEFSTR<letterlst>	S	Defines all variables beginning with the letters in the <letterlst> as integer numeric variables
DIM<VAR>(<num> <num>)[,<var> <num>],...]	S	Sets up maximum subscript values for arrays and makes available space for them
EDIT[<line1>]-[<line2>]	C	Transfers to TEXT program line, or lines, for editing
END	S	Terminates the BASIC program execution, closes all files, and returns you to command level
EOF(<filenum>)	S-I/O	End of file condition on file opened as <filenum> is indicated
ERL	S	Gives the line number where the last error occurred
ERR	S	Gives the error code number of the last error
ERROR<err num>	S	Simulates error number <err num>
EXP(<numex>)	F	Returns the number e raised to the <numex> power
FILES	C	Lists all RAM files on the LCD
FIX(<numex>)	F	Makes <numex> an integer by truncation
FOR<countvar> = <start val> TO <end val>[STEP <increment>]	S	Used with NEXT statement to repeat program lines a specified number of times
FRE<dummy numex>	F	Gives the number of unused bytes in RAM memory
GOSUB<linenum>	S	Used with the RETURN statement to execute and return from a subroutine beginning at <linenum>
GOTO<linenum>	S	Branches program execution to <linenum>
HIMEM	S	A system variable representing the highest RAM memory address available to BASIC. Its value is set by the CLEAR command
IF<expression> THEN<clause>[ELSE <clause>]	S	If <expression> is true, statement(s) in THEN <clause> are performed. Otherwise, performs the ELSE <clause> or goes to the next line if ELSE clause is omitted
INKEY\$	S	Checks for a character from the keyboard but does not wait for it
INP(<portnum>)	F	Reads a byte from <portnum>
INPUT [“<prompt>” ;]<variable list>	S-I/O	Reads data input from the keyboard into the <variable list>
INPUT#<filenum>, <variable list>	S-I/O	Reads data from file opened as <filenum> into <variable list>

BASIC Statements—Table of Formats, Descriptions

Statement	Type	Description
CDBL(<numex>)	F	<numex> converted to a double-precision number
CHR\$(<numex>)	F	Gives character with ASCII code <numex>
CINT(<numex>)	F	<numex> converts to integer by truncating decimal portion
CLEAR [<string space>], [<high mem>]	C	Program variables cleared. Optionally protects high memory
CLOAD ["<filename>"] [,R]	C	Loads a program from cassette into BASIC memory and optionally runs it
CLOADM ["<filename>"] [,A]	C	Loads a Machine language program file from cassette tape into memory
CLOAD?"<filename>"	C	Compares the program currently in BASIC memory with the copy on cassette it was loaded from
CLOSE [<file num list>]	S-I/O	Closes the files opened as filenum(s)
CLS	S-I/O	Clears the screen
COM <ON,OFF, or STOP>	S	Communications activity via the RS-232-C enabled or disabled
CONT	C	Continues program execution after a break or STOP
COS(<numex>)	F	Computes cosine of angle <numex>, <numex> is in radians
CSAVE"<filename>" [,A]	C	Save the contents of BASIC memory to cassette on tokenized BASIC or ASCII format
CSAVEM <startadd>, <endadd> [,<entryadd>]	C	Saves a Machine language program from "<filename>", to cassette tape
CSNG(<numex>)	F	<numex> is converted to a single-precision number
CSRLIN	S	Gives the vertical, line number, position of the cursor
DATA <constant list>	S-I/O	Data table is created to be read by READ statement
DATE\$	S	Holds the system date in the format "mo/dy/yr"
DAY\$ "<day>"	S	Sets or displays the system value for day of the week
DEFDBL<letterlst>	S	Defines all variables beginning with the letters in the <letterlst> as double precision numeric
DEFINT<letterlst>	S	Defines all variables beginning with the letters in the <letterlst> as integer numeric variables

BASIC Statements—Table of Formats, Descriptions

Statement	Type	Description
INPUT\$(<numex>)	F	Reads <numex> characters from the keyboard
INSTR ([<startpos>], <search str> , <match str>)	F	Returns the position of the first occurrence of <match str> in <search str> starting at <start pos>
INT(<numex>)	F	Gives greatest integer that is less than or equal to <numex>
IPL " <filename> "	S	Defines a BASIC program as the first to execute when the computer is powered on
KEY <num> , <string ex>	S	Function key <num> is set to the value of the <string ex>
KEY(<keynum>) <OFF,ON, or STOP>	S	Enables and disables trapping of function key <keynum>
KEY LIST	C	Displays on the LCD the current values of the function keys
KILL " <filename> "	C	Erases RAM file <filename>
LEFT\$(<string ex> , <length>)	F	Gives the leftmost <length> characters of <string ex>
LEN(<string ex>)	F	Returns the number of characters in <string ex>
LET <variable> = <expression>	S	Assigns the value in <expression> to <variable>
LINE([<x1> , <y1>]) - <x2> , <y2> [, <switch>] [,B[F]]	S-I/O	Draws a line between x and y coordinate pairs. B uses the coordinate to draw a box and F fills it
LINE INPUT [" <prompt> ";] <stringvariable>	S-I/O	Reads an entire line from the keyboard, ignoring commas or other delimiters
LINE INPUT# <filenum> , <stringvariable>	S-I/O	Reads an entire line from the file opened as <filenum>
LIST[<line1>] [- <line2>]]	C	List specified program line(s) to LCD screen
LLIST[<line1>] [- <line2>]]	C	List specified program line(s) to the printer
LOAD[" <device> :] [<filename or configuration> "] [,R]	C	Loads a program from a specified device and optionally runs it
LOADM " <dev:>] [<filename> "]	C	Loads a Machine language program from cassette or a RAM
LOG(<num var>)	F	Returns the logarithm to base e of <num var> (natural log)

BASIC Statements—Table of Formats, Descriptions

Statement	Type	Description
LPOS(<dummy num>)	F	Gives the carriage position of the printer
LPRINT<list of expressions>	S-I/O	Prints data on the printer
LPRINT USING<v\$>; <list of expressions>	S-I/O	Prints data on the printer using the format specified in <v\$>
MAXFILES	S	A system variable containing the maximum number of files a BASIC program may use
MAXRAM	S	A system constant equaling the highest RAM memory adress. Its value is 62960
MDM <ON/OFF/STOP>	C	Enables, disables, or holds the modem interrupt defined by an ON MDM GOTO
MENU	C	Exits BASIC and returns to the Main Menu
MERGE“[<dev>:] [<filename or configuration>]”	C	Merges the lines from an ASCII program file into the program currently in memory
MID\$(<string ex>, <start pos>[, <length>])	F	Returns <length> characters from the string <string ex> starting at position <start pos>
MOTOR <ON/OFF>	C	Turns an attached tape recorder on and off
NAME“[RAM:] <oldnam>” AS “[RAM:]<newnam>”	C	Changes the name of a RAM file
NEW	C	Deletes the current program in memory and clears all variables
NEXT[<countvar>]	S	Ends a FOR...NEXT loop
ON COM GOSUB <linenum>	S	Defines a trap subroutine for RS-232-C communications activity
ON ERROR GOTO <linenum>	S	Defines error trap routine to start at <linenum>
ON <numex> GOSUB <linenum>[<,line num>]	S	Exits to subroutine <linenum> specified by <numex> Used with RETURN statement
ON<numex> GOTO <lineNUM>[<,line num>...]	S	Exits to <linenum> specified by <numex>
ON KEY GOSUB <linenum>[<,line num>...]	S	Enables trap subroutine for specified function keys
ON MDM GOSUB <linenum>	S	Defines an interrupt subroutine to execute when modem activity occurs
ON TIME\$=“<time>” GOSUB <linenum>	S	Defines an interrupt subroutine to execute when the <time> value matches the system clock time

BASIC Statements—Table of Formats, Descriptions

Statement	Type	Description
OPEN"[<dev>:] <filename or configuration> FOR<mode>AS <filenum>	S-I/O	Opens a device to allow I/O as <filenum>
OUT<portnum>, <byte val>	S-I/O	Sends a byte to a CPV port
PEEK(<memory address>)	F	Reads the byte from <memory address>
POKE<memory address>,<byte val>	S-I/O	Places a byte in <memory address>
POS(n)	F	Gives the current cursor column position
PRINT<list of expressions>	S-I/O	Prints data on the screen
PRINT USING<v\$>; <list of expressions>	S-I/O	Prints data in <list of expressions> using format specified by <v\$>
PRINT#(<filenum>), (list of expressions)	S-I/O	Writes the (list of expressions) sequentially to a file
PRINT#(<filenum>), USING <v\$>;<list of expressions>	S-I/O	Writes the <list of expressions> sequentially to a file opened as <filenum> using the format specified by <v\$>
PRESET(<xcoor>, <ycoor>)	S-I/O	Turns off the specified LCO pixel
PSET(<xcoor>, <ycoor>)	S-I/O	Turns on the specified LCD pixel
POWER <numex>	C	Sets the duration of computer inactivity tolerated before an automatic power down
POWER CONT	C	Disables the automatic power down function
POWER OFF	C	Turns the Model 100 off from within an executing BASIC program
READ<variable> [,<variable>]	S-I/O	Reads data from a DATA statement and assigns them to variables
REM <remark>	S	Inserts remarks in a program
RESTORE <linenum>	S	Allows DATA statements to be reread from a specified <linenum>
RESUME [<linenum or NEXT>]	S	Continues program execution
RETURN <linenum>	S	Returns from a subroutine
RIGHT\$(<string ex>,<num>)	F	Gives the rightmost <num> characters of string <string ex>

BASIC Statements—Table of Formats, Descriptions

Statement	Type	Description
RND(<numex>)	F	Returns a pseudo random number between 0 and 1
RUN“[<dev:>] <filename or configuration>” [,R]	C	Executes a program The [,R] option is used to keep files open
RUN<linenum>[,R]	C	Executes a program at the <linenum> specified
RUNM“[<dev:>] [<filename>]”	C-I/O	Loads and runs a Machine language program
SAVE“[<dev:>” <filename or configuration> [,A]	C	Saves a BASIC program file to <dev:>. The [,A] option saves the program in ASCII format
SAVEM“[<dev:>] <filename>, <startadd>, <endadd> [,<entryadd>]”	C	Saves a Machine language program to RAM or cassette
SCREEN<ON or OFF>	S-I/O	Turns on and off the function log lable line in the LCD
SGN(<numex>)	F	Returns the sign of (<numex>)
SIN(<numex>)	F	Returns the trigonometric sine function of <numex>
SOUND<freq>, <duration>	S-I/O	Produces sound through the speaker
SPACE\$(<num>)	F	Gives a string constant of <num> spaces
SQR(<numex>)	F	Gives the square root of (<numex>)
STOP	S	Stops program execution and returns to command level
STR\$(<numex>)	F	Converts the value of <numex> to a string value
STRING\$ (<num>, <char>)	F	Returns a string of length <num> whose characters are all ASCII code of names or the first character of a string expression
TAB(<numex>)	F	Tabulates to a position <numex> spaces from current position
TAN(<numex>)	F	Gives the trigonometric tangent of (<numex>)
TIME\$ = “<hh>: <mm>:<ss>”	S	Sets or retrieves the system time
PRINT TIME\$	C	Prints current time
VAL(<string ex>)	F	Returns the numerical value of the string <string ex>
VARPTR (<var>)	F	Returns the address of a variable in memory

BASIC String Functions Table • Battery Backup

BASIC String Functions Table

String Function	Operation Performed
CHR\$	Gives character of ASCII code
INKEY\$	Gives keyboard key currently pressed
INPUT\$	Gives the characters from the keyboard
LEFT\$	Gives the left part of a string
RIGHT\$	Gives the right part of a string
SPACE\$	Gives a string constant of spaces
STR\$	Converts a number to a string value
STRING\$	Gives a Character string

BASIC Variable Names The rules governing variable names in BASIC are:

- must start with a letter;
- may contain any number of characters but only the first two are significant. BASIC cannot distinguish between variables named swim and sweet, for instance.
- cannot be a reserved word such as IF, ON, THEN, GOTO, etc.; or a reserved word followed by a type declaration character (\$,%,!,#); or a reserved word embedded within a variable name, as AND is embedded in RAND. See BASIC Reserved Words—Uses and Restrictions of.
- must end with a type declaration character. The default value for the type of variable is double-precision numeric, so any variable name not ending in \$, !, or # is automatically a double-precision numeric integer variable. The type declaration characters are: \$ for a string variable (for 0 to 255 letters, numbers, punctuation marks, and other characters); % for numeric integers (whole numbers from -32768 to +32767); ! for single-precision (numbers with decimal fractions and six significant digits); # for double-precision numbers (numbers with decimal fractions and fourteen significant digits.) Exponential notation is allowed for both double- and single-precision variables. In single-precision variables using exponential notation, the exponent should be prefaced with an E. Double-precision exponents are prefaced with a D.

An exception is any variable which begins with a series of characters specified in one of the type declaration statements:

DEFINT—for integer numbers, DEFSNG—for single-

precision numbers, DEFDBL—for double-precision numbers, DEFSTR—for string variables.

For example, if you place the statement, DEFSTR A,B,N-P at the beginning of your program, all variables starting with A,B,N,O, and P will be string variables because they start with one of the letters defined in the DEFSTR to be a string variable prefix.

BASIC Words, Reserved See BASIC Reserved Words, Uses and Restrictions of.

Batch Processing Running a program or series of programs with no interaction between the user and the program.

Batteries The Model 100 operates on power from either an AC to DC converter plugged into a wall socket, or four AA alkaline batteries, which give true portability. Radio Shack recommends that you use their brand (RS-23-552). If the batteries are installed, and you are connected to the power adapter unit, even if the power source is turned off, the Model 100 will not use the batteries.

The estimated lifetime of one set of four AA batteries is 20 hours of operation. This means twenty one-hour sessions, one twenty-four hour session, or anywhere in between. When the batteries have twenty minutes or less of power remaining and you turn on the computer under battery power, the low battery indicator light, located just below the Radio Shack label, lights up. You must replace the batteries or plug in the AC/DC adapter if you want to use the computer once the batteries are dead, however the RAM memory will continue to be powered by the built-in ni-cad (nickel-cadmium) battery for another eight to thirty days depending upon the size RAM you have. The larger the RAM memory the more quickly it will drain the built-in ni-cad battery. Loss of RAM memory is not likely, however, if you use the computer again before the ni-cad is drained, because powering on with a new set of batteries or plugging into wall current will recharge the ni-cad battery.

Battery A device which produces electrical energy by chemical means. The Model 100 is optionally powered by four AA alkaline batteries which produce six volts direct current (DC).

Battery Backup During a power failure, batteries supply auxiliary power to the processor, so volatile information is not lost. The Model 100 contains a rechargeable, built-in ni-cad (nickel-cadmium) battery to back up RAM memory when the AA alkaline batteries wear out or when wall current, via the adapter/converter, is disconnected.

Baud • Bi-Directional

Baud The binary units of information transmitted per second or the number of bits transmitted per second. The Model 100 built-in modem operates at 300 baud. The RS-232 C interface can transmit data at rates ranging from 300 to 19200.

Baud Rate The measurement of data flow in bits-per-second. The TELCOM default value is M,

which signifies the modem will be used. The modem operates at a baud rate of 300 bits-per-second. If you change the baud rate from M to some other value, even 300, the modem will be disabled and you will be setting the baud rate for the RS-232C interface. Other acceptable values for the baud rate follow:

TELCOM Value	BAUD Rate	Device	Used With
M	300	modem	
1	75	RS-232C	rarely used
2	110	RS-232C	do teletypewriters
3	300	RS-232C	slow-speed serial printer
4	600	RS-232C	printers and disc devices
5	1200	RS-232C	high speed serial printers, some modems
6	2400	RS-232C	direct computer to computer
7	4800	RS-232C	"
8	9600	RS-232C	"
9	19200	RS-232C	"

Baud Rate Generator An oscillator, usually adjustable, providing clock signals for the connection of a peripheral. Baud rates in the Model 100 range from 300 to 19200 baud.

Baudot An older communications code, named for the man who invented it, and used for five-level (hole) teletypewriter and telex machines. Other codes used are ASCII and EBCDIC, which are eight-level codes.

Bay Area Systems Engineering BASE. Originators of the *TRS-80 User's Encyclopedia (Model 100 and NEC PC)* and other information resources for the personal computer user.

B-Bus The second source-bus to the arithmetic logical unit in a two-or three-bus processor.

BCD Binary Coded Decimal. A four-bit binary representation of the ten decimal digits 0 through 9. Six out of the sixteen possible codes are unused requiring the use of a "Decimal Adjust" instruction for correct binary addition. 1 is encoded as 0001, 9 as 1001. Two BCD digits are usually packed in a byte.

BCP Byte Control Protocol. A protocol for communications between two computers or devices

which use a special character to identify the start of a message which includes both a count of the number of data bytes, and the actual data bytes. Also called byte count oriented protocol.

BEEP BASIC Statement. Causes the computer to emit a high pitched "beep" from its sound generator for a duration of about one-half second. This is the same sound the computer sometimes makes in response to an error condition.

Bell Laboratories Research laboratories in New Jersey that have made many major advances in the electronic and computer fields.

Benchmark Program A specific program written to calibrate the speed of a computer in a well defined situation, or type of computation, e.g. scientific "number crunching," sorting, or compilation.

Beta Test Site vs. Alpha Test Site See Alpha and Beta Test Sites.

Bi-Directional Data flow may go in either direction on a wire. At each end of the wire there are transceivers to both receive and transmit. Common bi-directional buses are the tri-state bus and the open collector transistor-transistor logic bus. The

Model 100 modem is another example of a bi-directional device.

Bi-Directional Printing Alternately printing in either direction. A line printed left to right is followed by a line printed right to left, avoiding a carriage return delay and greatly increasing the printing rate.

Binary Counter An electronic device which outputs a sequence of ascending or descending binary numbers.

Binary Number A representation of an integer as a sum of powers of 2, using a sequence of 0s and 1s.

Binary Search A technique where the search is divided by two at every interval.

BIOS A Basic Input/Output System. Manages serial peripherals. The BIOS is often ROM resident.

Bipolar A technology of integrated circuit fabrication which uses transistor switching elements based on majority carriers for switching and amplification.

Bistable A device that is always in one of two possible stable states.

Bistable Multivibrator Flip-flop. Active elements able to assume either one or another stable states characterize a flip-flop circuit.

BISYNC (Pronounced "by-sink") Binary SYNchronous Communications protocol. An IBM character-oriented protocol for synchronous transmission of binary-coded data between two devices that uses a defined set and sequence of control characters.

Bit Binary digIT. A bit is a 0 or a 1. Bits are used in computer systems to code information, instructions, and data. Larger units of bits are: nibbles (4), bytes (8), or words (16, 24, 32, 96, or more).

Bi-Tech Bar Code Reader* Reads all common bar codes. Features a push-to-read optical sensor consisting of a 700nm light source, precision aspheric optics, integrated circuit detector, and internal metal shielding. Hardware Accessory. B.T. Enterprises.

Bit-Parallel Data transmission method where every digit of a binary number is sent over a separate wire simultaneously. The Model 100 printer interface is a Centronics standard bit parallel interface.

Bit-Slice A vertical slice of a computer. This component constitutes an n-bit slice of a traditional CPU, minus control. Usually $n = 4$. A bit-slice implements a complete data path across the CPU, including multiplexers, ALU, shifters, registers, and

accumulators. A typical case would be two four-bit, bit-slice processors linked together to make up one eight-bit processor.

Blank Line On Printer, Print An LPRINT statement entered from BASIC with no other specifications will print a blank line (that is, feed the paper up one line and return to left margin) so you can space your printout format neatly.

Blanking On the LCD screen, not displaying a character, leaving a space. If you use an auto log-on sequence in TELCOM, you will see the phone number and your ID code display, your password will not appear on the screen.

Block Within a logical record, information is stored in units or blocks. Block is also sometimes used to mean a collection of logical records, as in blocked records or blocking factor. Block size is usually expressed in bytes. Text mode allows you to define blocks of text and move, copy, and delete them as a single unit.

Block Manipulations of Text Manipulating Text Blocks. To delete or move a block, use F6, the CUT key. To leave the block intact after copying it elsewhere, use F5, the COPY key. In either case, the marked block will be copied to an internal storage location, called the paste buffer. If you chose the CUT (F6) option, the text below will move up to fill in the space the block used to occupy. In either case you can now insert the block of text you have in the paste buffer at any place in your text file or even into another file. All you need to do is place the cursor in the file where you want the insert to go and press the PASTE key. The text after the cursor will move over and down to make room. Once you place a block of text in the paste buffer, it will remain there until you replace it with another block or cold start the computer. This allows you to make multiple copies of a block.

Block Mark in Text Defining Text Blocks. If massive rewrites are necessary, changing the file one character at a time can be pretty tedious. Fortunately, TEXT mode allows you to define a segment of the text as a block so that you can manipulate the block as a whole. You can then erase, move, or copy the block. To define any segment of text as a block, position the cursor over the first character and press F7, the SELECT key in TEXT. Next, use the cursor movement keys to move the cursor to the last character you want in the block. All the text in between will appear in inverse video, grey on black, rather than black on grey, to indicate that it is now marked as a block.

Block of Memory • BREAK Program Execution

Block of Memory A series of consecutively numbered bytes in a device (usually internal). See Address.

Blockade* The more targets you hit, the longer your snake grows. Keep an eye out for walls and other obstacles. Play against the computer or with a friend. Three levels of difficulty. 24K; cassette. SilverWare.

BNPF Representation An older data encoding format for PROM programmers using the characters B = beginning, F = finish, N = negative (1), P = positive (0). For example: the byte 10010110 is represented as BNPPNPNPF (pronounced "Smith.")

Board, Breadboard The fiberglass or pressed paper sheet used for mounting the integrated circuits. Interconnections may be wire-wrapped, soldered, or printed on the board. The term breadboard refers to a prototype circuit and dates from the time when radios were made on mother's breadboard. Also called a card when referring to smaller boards that plug into the motherboard.

Boards/Cards/Interfaces See 8K Memory Models.

Board-Tester A computer controlled device that performs electronic tests on printed circuit boards.

Books for the Model 100 The following is a list of books for the Model 100.

The TRS-80 Model 100 Portable 100:

A complete step-by-step learner's manual

David A. Lien

Compusoft Publishing

535 Broadway, Dept 171083

El Cajon, CA 92021

800-854-6505

How to Do It on the TRS-80 Model 100:

for the Model I,II,III, Color Computer, and Model 100

William Barden, Jr.

IJG, Inc.

1953 West 11th St.

Upland, CA 91786

714-946-5805

TRS-80 Model 100 Hot Sheet

G. Camp and P. Wiener

IJG, Inc.

1953 West 11th St.

Upland, CA 91786

714-946-5805

44 Programs for the Model 100 Portable Computer

96 pages of practical programs for business and the home.

Jim Cole

ARCsoft Publishers

Directory of Online Databases

Your guide to 1600 on-line databases.

Cuadra Associates, Inc.

2001 Wilshire Blvd., Suite 305

Santa Monica, CA 90403

213-829-9972

Telex 652421 CUADRA SNM

Boolean Logic Named after George Boole, who defined an algebra of logical operations such as And, Or, and Not, on the two values true and false. See BASIC Logical Generator Table.

Boot To use a bootstrap. Generally used to describe starting up a computer.

Bootstrap A program used for starting the computer, that usually clears memory, sets up I/O devices, and loads the operating system from ROM, diskette, or cassette.

BOP Bit-Oriented Protocol. A protocol for communications between two computers or devices which causes a special bit pattern to separate groups of data bits. The RS-232-C interface and modem translates the bytes, 8 parallel bits, which are used internally by the Model 100 to a serial data stream of single bits, used for communication between computers. This data stream contains special bits separating the data bits which make up each byte when translated back to parallel form. Generally these special bits are start bit and stop bit, bracketing each set of 8 data bits in the serial communications stream.

Bouncing Short intermittent conduction from vibration of switch contracts after closure. Usually present in keyboard input and eliminated by special hardware or software (debouncing).

Bound Processor-bound or I/O-bound, indicating which component of a system is preventing faster performance.

Bowling League Secretary* Keep the team stats. 24K. Chattanooga Choo Choo Software (for Prickly-Pear Software).

BPI Bits Per Inch. Used to specify the density of data recorded on tape or disk.

Branch A programming instruction which causes transfer of control to another program sequence. In BASIC, three control commands may be used: GOTO, GOSUB, and CALL. The first two branch execution to another, non-sequential line number in the same BASIC program, and CALL branches to a Machine language subroutine. See also ON...GOTO, ON...GOSUB, BASIC Interrupt Commands.

BREAK Program Execution You may interrupt BASIC program execution by pressing the SHIFT

and BREAK/PAUSE keys together. The CONT command will resume program execution when keyed and ENTERed from the keyboard. While the program is paused in this way you can examine variable contents by PRINTing them and even assign new variable values. CONT will not resume program execution if you alter any BASIC program lines. Using END as a program line will also cause a break in execution to the same effect.

BREAK/PAUSE Key The BREAK/PAUSE key temporarily pauses current BASIC program execution when pressed once and resumes it from the current line when pressed a second time. This is useful if, for instance, you want to pause while listing a BASIC program so that you can read it before it scrolls off the screen. Pressing SHIFT and BREAK/PAUSE at the same time breaks program execution. BASIC programs display the message break in <xxxx> where <xxxx> is the program line that was executing at the time of the BREAK. Typing and ENTERing the BASIC keyword CONT resumes program execution at the line where the interrupt occurred. In most other applications, such as printing a TEXT file, pressing SHIFT and BREAK/PAUSE will stop executing the current function, but there is no way to resume it without restarting the function from the beginning again. In such a case the message aborted is displayed on the LCD. Usually pressing the BREAK/PAUSE key alone has no effect.

Breakpoint A point at which the processor stops a program sequence and displays the current machine status, implemented through hardware, software, or a combination of both. See STOP.

BTAM Basic Telecommunications Access Method (IBM term used on mainframes).

Bubble Memory Memory utilizing microscopic magnetic domains in an aluminum garnet substrate. Present memories have 92K bits per device. Future devices should boast better than one million bit storage density per chip.

Buffer In software, any memory structure provided for the temporary storage of data. In hardware, a device which restores logic drive signal levels in order to drive a bus or a large number of inputs.

Buffering The delaying and temporary storage of data in a data communications path.

Bug Errors in a Program. A programmer must insure that a program correctly processes all of the types of data it is intended for. Samples of the data are prepared (test data) and the program is exe-

cuted using this data (a test run). The program's outputs (reports, screen displays, files, etc.) are then verified to be as specified. An error in the processing logic of a program is called a "bug," hence the term "debug" and "bug-free."

Bulk Storage Large capacity data storage, generally long term.

Burn-In A phase of component testing where basic flaws or early failures are screened out by running the circuit for a specified length of time, generally at elevated temperatures in some sort of oven.

Bus Path for signals having a common function. Every Standard MPU creates three buses: the data bus, the address bus, and the control bus.

Bus Controller The unit in charge of generating bus commands and control signals.

Bus Extender A device which allows additional cards to be plugged into a computer's bus. Also known as an expansion chassis.

Bus Termination An electrical means of preventing reflections at the end of a bus that is only necessary in high-speed systems or some specially designed low-speed systems.

Business Application Software See Business Manager Series, The; BusinessPak+; EXPNS+; PortaStat; PUT+; Traveling Accountant, The; Traveling Appointment Manager, The; Traveling Expense Manager, The; Traveling Project Manager, The; Traveling Project Manager, The; Traveling Sales Manager, The; Traveling Tax Manager, The; Traveling Time Manager, The.

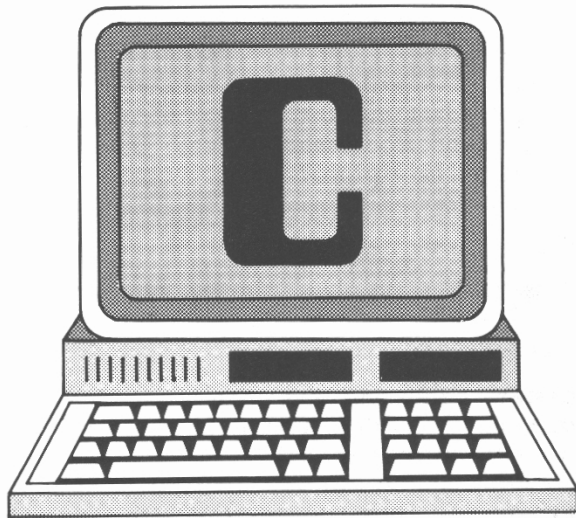
Business Manager Series, The* An eight-program package (each sold separately) designed for business persons who spend much of their time on the road. These programs aid in organizing and recording information concerning sales, expenses, work completed, and task completion time. Any of the accumulated information can be transmitted back to office computer via phone lines or direct cable connection. See Traveling Accountant, The; Traveling Appointment Manager, The; Traveling Communicator, The; Traveling Expense Manager, The; Traveling Project Manager, The; Traveling Sales Manager, The; Traveling Tax Manager, The; and Traveling Time Manager, The. Cassette. Traveling Software, Inc.

BusinessPak+* A six-program package to aid business persons to write, organize, calculate, and manipulate important data for personal use or presentations. Each program is loaded on its own

Byte

cassette along with any other files that are needed for execution. An excellent instruction manual is included which provides detailed documentation for each program. See EXPNS+*, GRAPH+*, PUT+*, SORT+*, TELEX+*, WRITE+*, DATA+*, and SORT2+*. 16K; cassette. Portable Computer Support Group.

Byte A group of 8 bits. A byte is universally used to represent a character. Microcomputer instructions generally require one, two, or three bytes. One byte has two nibbles (4 bits). The Model 100 uses a basic unit of byte. The bytes used to store variable data differ by variable type. Single-precision numeric variables use four bytes, double-precision numeric eight bytes, integer variables use two bytes, and string variables use one byte per character.



C Codes. ASCII 67, HEX 43. c—ASCII 99, HEX 63

C Symbol for the carry bit; the clock; and the hexadecimal notation for the decimal number 12: C base 16 = 12 base 10 = 14 base 8 = 1100 base 2.

C Setting Printer Form Length. To set form length on the Epson RX-80 printer in lines- or inches-per-form or page, either enter the BASIC statement:

`LPRINT CHR$(27); ("c");CHR$(<n>)`

<n> is the number of lines for the form length and must be between 1 and 127, or enter

`LPRINT CHR$(27); ("c");CHR$(0);(CHR$(<n>))`

<n> is the number of inches for the form length and must be between 1 and 22.

C A high-level programming language developed at Bell Laboratories, associated with the UNIX operating system.

C 80C85 Assembly Language Instruction. Condition Call. If the specified condition is true, the action specified in the CALL instruction is performed; otherwise, control continues sequentially. The addressing mode is immediate or register indirect. No flags are set. See CALL.

Cables and Adapters Acoustic Coupler (RS-26-3805), CCR-81 Recorder-to-Computer Connection Cable (RS-26-1207), Model 100 Modem Cable (RS-26-1410), Model 100 Parallel Printer Cable (RS-26-1409), Null Modem Adapter (RS-26-1496), RS-232C Cable (RS-26-4403), TRS-80 AC Adapter (RS-26-3804).

Cables, Installing Cassette See Cassette Cable, Installing.

Cables, Installing Modem See Modem Cable, Installing.

Cables, Installing Printer See Printer Cable, Installing.

Cache Used between the central processor and main memory, a cache is a high-speed buffer. Filled at medium speed from main memory, instructions and programs can operate at high speed if found in the cache. A new instruction sequence is loaded from main memory if not found in the cache.

CAD Computer-Aided Design.

CAI Computer-Assisted Instruction.

CALL BASIC Statement. Allows a Machine language subroutine, starting at a given address in memory, to be accessed through a BASIC program. The format is:

`CALL <address>[,<variable>[,<variable>]]`

<address> must be a numeric variable. Its value is the starting memory address of the Machine language subroutine being called.

<variable> is the name of a variable which is to be passed as an argument to the Machine language subroutine. In the Model 100 the first variable is passed to the A register, and may range in value between 0 and 225. The second variable is passed to the HL register. Acceptable values for the second variable are -32768 and 65535 inclusive. Both variables are optional. See Argument.

Call Program execution is temporarily transferred to a subroutine or subprogram. When completed, execution resumes at the instruction following the call.

CALL 80C85 Assembly Language Instruction. The high-order bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction. The addressing mode is immediate or register indirect. No flags are set.

Call BASIC Subroutine See GOSUB and RETURN, RETURN ON MDM GOSUB, ON COM GOSUB, ON TIME\$, and ON GOSUB.

Call by Reference The storage locations of the parameters are passed to the subroutine by the CALL, rather than passing a copy of the values.

Call by Value A subroutine or procedure call in which the actual values of the parameters are

CAM • Cassette File, LOAD Instructions for BASIC

passed to the subroutine, such as in the BASIC CALL statement.

CAM Content Addressable Memory. Associative memory, addressed by contents rather than location.

CAMAC Computer Automated Measurement And Control. An application where, based on measurements taken, a computer automatically measures a process and controls one or more devices affecting the process. Also called process control.

Cancel Current Line Press the SHIFT and BREAK/PAUSE keys together. This is valid only in BASIC. The canceled line will remain in the LCD followed by a ^C, but will not appear if you list the program.

Cancel the Current Operation When the SHIFT and BREAK/PAUSE keys are pressed together the current operation is canceled.

Caps Lock Key The key that locks the keyboard in upper case mode and toggles on and off with each press. When on, it causes each alpha key press to generate the letter in upper case. When off, pressing the SHIFT and an alpha key together generates an upper case letter. An alpha key press alone generates the letter in lower case. The caps lock key does not "shift" non-alpha keys. This must be done with SHIFT.

Card Cage A rack which supports the printed-circuit boards in a computer.

Carriage Position, Printer BASIC. See LPOS.

Carriage Return A standard typewriter key which causes the printing element to move back to the beginning of the line. To move the paper up, a separate line feed must be added. Frequently, the microprocessor interprets a carriage return to mean end of line or end of command.

Carriage Return Press ENTER to end the current line and give the line to the requesting program. This is valid under BASIC and all modes, except TEXT, where pressing ENTER indicates the end of a text block or paragraph.

Carrier A frequency used to "carry" information, which is modulated to denote 0 or 1. When you set the ANS/ORIG switch on the left side panel to ANS, the carrier frequency for the modem modulates between 2225 and 2025 Hz. When you set to ORIG, the carrier frequency modulates between 1270 and 1070 Hz. See Modem.

Carry In the status register of the central processor, a carry is a flag bit which indicates an operation

overflow by the arithmetic logic unit. The carry is also used during shifts.

Carry Look-Ahead A circuit which predicts, from partial adders, the final carry from an addition. Binary addition is speeded up significantly by eliminating the carry propagation delay.

CAS Column Address Strobe. Used for addressing in dynamic memory control.

Cassette A small, plastic cartridge containing two spools of 1/8" magnetic tape which must meet standards for digital recording.

Cassette Cable, Installing Before saving files on cassette, the Model 100 must, of course, be connected with a cassette recorder. Radio Shack recommends that you use their CCR-81 Computer Recorder (RS-26-1208) which comes with its own connector cable (RS-26-1207). The connector cables are the same as the ones used by the other TRS-80 machines. Before hooking up a cassette recorder, or any other peripheral device, it is a good idea to turn the computer off. The cassette cable consists of a large cylindrical plug on one end and three smaller plugs on the other, two of which are grey and the third black. The large cylindrical plug inserts into the leftmost socket on the rear panel of the computer, marked Cassette. Of the three smaller plugs on the other end of the cassette connector cable, the black one plugs into the EAR connector of the cassette recorder, the smaller of the two grey plugs inserts into the REM connector on the recorder, and the larger grey plug inserts into the AUX connector on the recorder. Having made these four connections, one to the computer and three to the tape recorder, you may turn the Model 100 back on, turn on the cassette recorder, and you're ready for the cassette I/O.

Cassette Directory See CLOAD.

Cassette File, LOAD Instructions for BASIC To LOAD a BASIC program SAVED on cassette, enter the BASIC mode.

For a program on cassette named "SAMPLE," hook up the cassette recorder, rewind the cassette, press PLAY and enter:

LOAD "CAS:SAMPLE"
or
CLOAD "SAMPLE"

You may also press F2, the LOAD key in BASIC, wait for the prompt LOAD" and then enter CAS:SAMPLE. In each case, if you omit the filename, BASIC will LOAD the first BASIC file found on the cassette tape. If you do specify the filename, BASIC will

Cassette File Loading—Troubleshooting • Cassette Save Instructions—TEXT File

print the messages SKIP:<filename> for each file found on the tape that is not SAMPLE, and FOUND: SAMPLE when it finds your program.

To automatically run the program after it is LOAded, use LOAD "CAS:SAMPLE",R or its synonym CLOAD "SAMPLE",R. See also RUN.

LOAD instructions erase all program lines in memory before the LOAD. To combine a SAVED program with the one you are writing, see MERGE.

Cassette File Loading—Troubleshooting Normally you will not encounter problems when loading files from cassette tape. If you do encounter difficulties, however, here are several measures to try:

1. Check the volume setting on the cassette recorder. It should be set between 4 and 6 for best results.
2. Be sure the tape is rewound sufficiently to position it before the beginning of the file. Rewind to the beginning of the tape if you are not sure of the start location of the file. Then be sure to specify a file name in your LOAD, CLOAD statements or BASIC will load the first file it finds on the tape. To rewind the tape you need to disconnect the cassette plug from the REM connector. This returns control of the cassette motor to the cassette recorder.
3. Be sure you are using a good quality tape, such as a Radio Shack certified tape, e.g. RS-26-301 or RS-26-302. Even one thin spot in the magnetic coating of your cassette tape may be sufficient to cause the loss of valuable bits of binary code. Certified tapes have been examined to insure that they contain no such thin spots.
4. Finally you may be experiencing interference from a nearby monitor or television. Try moving the cassette recorder to a location further away from the monitor.

Cassette File, SAVE Instructions for BASIC To save a BASIC program named SAMPLE on cassette you must be in the BASIC mode with SAMPLE in BASIC memory (ENTER BASIC and write SAMPLE) or SAMPLE should be a program already written and SAVED to RAM. If SAMPLE is currently a RAM file, you have two options for getting it into BASIC memory.

Position the cursor over the RAM file named SAMPLE.BA on the Main Menu and Press ENTER, which causes the program to autorun. Then, either wait for the program to finish executing or break program execution by pressing the SHIFT and BREAK/PAUSE keys together. In either case, this leaves you in BASIC command mode with the program SAMPLE in BASIC memory.

Alternatively, you may position the cursor over BASIC on the Main Menu and press ENTER. This places you in BASIC but you still need to LOAD the RAM file SAMPLE.BA into BASIC. To do this, enter LOAD "SAMPLE" and press ENTER. Because the device has been omitted from the LOAD command BASIC loads SAMPLE from the default device which is RAM. You may optionally press F2, the Load key in BASIC which automatically types the LOAD" portion of the load command for you, so you need only enter the name SAMPLE directly from the keyboard. Whichever means you use the result will be the same, you will end up in BASIC mode with the program SAMPLE in BASIC memory. Connect the recorder to the computer and press the RECORD and PLAY buttons together, locking them. To save SAMPLE to cassette, enter:

CSAVE "SAMPLE"

or

SAVE "CAS:SAMPLE"

BASIC automatically keys "SAVE" when you press F3, while you must enter the remainder manually, CAS:SAMPLE". In this case, you specify the non-default value, CAS:.

When you press ENTER, the recorder starts recording SAMPLE to cassette, stopping when the file has been saved. See Cables Cassette, Installing.

Cassette I/O BASIC Command Table

Cassette I/O Commands	Function Performed
CLOAD	Loads a tokenized BASIC format program from cassette
CLOAD?	Compares a cassette copy of a file with the copy loaded to BASIC
CLOADM	Loads a Machine language format file from cassette
CSAVE	Saves the current contents of BASIC memory to cassette
CSAVEM	Saves a Machine language program from internal memory to cassette tape
MOTOR	Turns on and off the cassette tape motor

Cassette Load, Verify See CLOAD.

Cassette Motor Control BASIC. See MOTOR.

Cassette Save Instructions—TEXT File Although your text file will be automatically saved as a RAM file when you exit TEXT, by pressing F8, the Menu key, or the ESC key twice, you may want to make a

Cassette TEXT File, LOAD • Change Contents of a File

copy off line on cassette tape. You can do this while in TEXT by pressing F3, the SAVE key. When "Save to" appears in the lower left corner of the screen, you type and enter a file name. When the copy is made the "Save to" prompt disappears. See TEXT.

Cassette TEXT File, LOAD To load a BASIC text file from cassette in TEXT mode, go to the beginning of the file you want or to the beginning of the tape if you don't know the file's location. In either case enter the TEXT mode file where the incoming data is to be stored and press F2. When the PROMPT LOAD FROM appears at the lower left corner, enter the file name. As each file on cassette is found, the message "SKIP" appears with the file name if it does not match the file you want to load. FOUND and the file name appears when the file is located. In this case, the file name is the same as the one given for the LOAD FROM prompt. When the file is loaded, the prompts disappear and you are ready to begin. See TEXT.

Cat's Paw Cribbage Challenger* Play against the computer or another person and let the computer keep score. 24K. L/R Software.

CATV Cable Television

CBASIC A popular BASIC language compiler for 8080, Z80, 8085, 80C85 and 8086/8088 microcomputers. Most BASICS are interpreted rather than compiled.

CCD Charge Coupled Device. A storage technology using metal-oxide semiconductor capacitors. It transfers charge from one cell to another in a recirculating fashion.

CDBL BASIC Function. CDBL converts any numeric expression <numex> to a double-precision number. The format is:

<variable> = CDBL(<numex>)

BASIC converts both integer and single-precision numeric values to double-precision by adding zero digits to the right of the decimal so that the number of significant digits are fourteen.

CDC Control Data Corporation. A manufacturer of large computer systems, peripherals, and disks.

CDILP Ceramic Dual In-Line Package. See DIP.

Cell A repeated unit in a RAM chip storing one unit of information and returning it in response to a particular address signal.

CERDIP (Pronounced "sir-dip") CERamic Dual-In-line Package.

Chaining A method of allowing the execution of programs larger than the main memory of a computer by loading and executing modules of the same program sequentially.

Change a BASIC line If you have not entered the current line to BASIC by pressing the ENTER key you may change it by backspacing, erasing the characters you backspace over, and then re-keying the portion of the line you erased.

If you have entered the line to BASIC you may replace the line by entering a new line with the same line number.

Alternatively, you may transfer the line, range of lines, or entire program to TEXT mode using the BASIC EDIT command. Then you may use the text processing abilities in TEXT to modify program lines. Pressing F8, the Menu key in TEXT, transfers these lines back to BASIC memory. See BASIC Editing.

Change Contents of a File Edit means to change the contents of a file—including both program files and data files.

The Model 100 stores three kinds of files in RAM memory. The operating system assigns one of three extensions depending upon the type of file it is. Extensions and file types are:

.DO—Document file, stored in ASCII format.

.BA—BASIC file, stored in encoded or tokenized format.

.CO—Machine language file.

The Model 100 text editor, TEXT, will only edit ASCII format files. However, you may temporarily translate BASIC files that are in BASIC memory to ASCII format using the BASIC EDIT command.

To edit a RAM document file, .DO, place the cursor on the Main Menu over the name of the file to edit and press ENTER. This displays the first eight lines of the file on the LCD and enables TEXT mode. Alternatively, place the cursor on the Main Menu over TEXT and press ENTER. When you see the prompt "File to edit?" on the LCD, enter the name of the document file in RAM you want to edit and press ENTER. The .DO extension is optional. This also displays the first eight lines of the file on the LCD and enables TEXT. You may now use all the text editing functions of TEXT mode on the file. When you finish editing, press F8, the Menu key in TEXT. This saves your ASCII file, with all the changes you have made in RAM, using the same file name you entered to begin editing, and returns you to the Main Menu.

To edit a RAM BASIC file, .BA, place the cursor on the Main Menu over the name of the file to edit

and press ENTER. This automatically executes, or runs, the BASIC program. Break program execution by pressing the SHIFT and BREAK/PAUSE keys at the same time. You will see the prompt, OK, indicating that you are in BASIC command mode. The BASIC file you entered will now be in BASIC memory as you can see if you key in and enter LIST. (The program lines from the BASIC file will list on the LCD.)

Alternately, you may place the cursor on the Main Menu over BASIC and press ENTER. You will see the prompt OK, indicating that you are in BASIC command mode. To get the BASIC RAM file to edit into BASIC memory press F2, the Load key in BASIC. When you see the prompt LOAD" on the LCD, key and enter the name of the file to edit. Otherwise, simply key and enter LOAD" and the file name. Your BASIC file to edit will now be in BASIC memory, as you will see if you key and enter LIST. (The program lines will list on the LCD).

To translate the file in BASIC memory into ASCII format for editing, enter the BASIC command EDIT. This translates your file, placing it into a separate, temporary RAM file. The first eight lines of your BASIC file on the LCD and TEXT mode will be enabled, allowing you to edit on your BASIC program.

When you are finished editing the BASIC program, press F8, the Exit key. This translates the program lines in your temporary edit-file from ASCII format back to tokenized BASIC form in BASIC memory. There is no need to save the program back to RAM. You can check this by exiting BASIC, pressing F8, which erases BASIC memory contents, and reentering your RAM program file into BASIC memory using one of the above methods. When you list the file you will see that your editing changes are there. See Edit TEXT Mode.

Change Date in System See DATE\$.

Change Memory Byte with BASIC See POKE.

Change Name of a File See NAME...AS.

Change Time in System See TIME\$.

Channel Logical connection from a CPU to an I/O device. To optimize communications, channels may be multiplexed, or have dedicated ports or a dedicated channel processor. See also DMA.

Character—Read from Keyboard BASIC. See INPUT, INPUT\$, IN-KEY\$, LINE INPUT, Keyboard, and I/O.

Character, Insert BASIC. In BASIC command mode you can only change the line you are in the process

of typing. Let us assume you have not yet entered the current line into BASIC memory by pressing the ENTER key. To edit the line backspace the cursor. This erases the characters you backspace the cursor over. Once you have backspaced over the mistake, or to the place in the line where you want to insert characters, you may retype the line as you want it to appear.

If you have already entered the line to BASIC memory you have two options. You may rekey the entire line, using the same line number and inserting the character you left out the last time, and then enter the new line to BASIC memory where it will replace the old line of the same number. Otherwise, key EDIT and the number of the line with the missing character and press ENTER. This translates the line to TEXT mode where you may insert the character according to the TEXT editing procedure. Then press F8, the EXIT key, to return the line to BASIC memory in its edited form where it replaces the old version of the same line.

Character, Insert TEXT Mode. Simply use the cursor control keys to position the cursor on the character that should go to the immediate right of the new character you will insert. Now, as you key new characters they will be inserted at the current cursor position while all characters to the right of the cursor move to the right to make room.

Character Delete BASIC. If you have not yet pressed ENTER to give the line to BASIC memory, backspace, erasing the characters you backspace over, until you have deleted the unwanted character. Then, rekey the portion of the line you want erased while backspacing.

If you have already entered the line to BASIC memory, you have two options. You may retype the entire line correctly, using the same line number, and then enter the line to BASIC memory where it will replace the old line of the same number.

Otherwise, type EDIT and the number of the line containing the character you want to delete and press ENTER. This translates the line to TEXT where you may use the cursor keys to place the cursor to the immediate right of the character to delete and press the DEL/BKSP key. This deletes the character from the program line. Then press F8, the Exit key, to return the line to BASIC memory in its edited form where it replaces the old version of the same line.

Character Delete TEXT. Simply use the cursor control keys to position the cursor over the character to delete and press the DEL/BKSP and SHIFT

Character File • CHR\$

keys at once. Alternatively place the cursor to the immediate right of the character to delete and press the DEL/BKSP key alone. In either case the character will be deleted and the remaining text will move in to close the gap.

Character File A file containing character data, letters, numbers, or special characters. Also called a text file. These files have a file extension of .DO for DObject. By contrast, an object file, such as .CO file, may contain data which is not displayable as characters. Most data and programs that you write will be in text files. See Data Files.

Character Generator A circuit which forms the letters or numbers on a screen or printer.

Character Set The collection of characters available for display or processing on a particular computer or peripheral. The Model 100 character set includes all 128 7-bit standard ASCII characters as 128 8-bit special graphics characters. See ASCII, ASCII Character Code Tables, CHR\$, CODE Key, and GRAPH Key.

Character String A one-dimensional array or sequence of characters, encoded as bytes. Character strings have a length field or are terminated by the zero byte. See Type Declaration Character, Type Declaration Statement.

Characters, Entering ASCII Control You can enter any possible ASCII character from 0 to 255 from the Model 100 keyboard using one to three keys. All ASCII characters 33-255 will display on the LCD. ASCII characters 0-32 do not display. These are the control characters. Most of the control characters may be generated from the keyboard by pressing an alpha key and the CTRL key together. The remainder are generated by the BREAK/PAUSE and ESC keys, the spacebar, and the four cursor control keys.

Although you won't see the control characters you generate on the keyboard you may get other results, such as the cursor jumping to another position or messages appearing onscreen, depending upon the application program you are using at the time. This is because the program receiving the ASCII characters you key is watching for certain keys to be pressed to indicate various actions other than data entry. If you generate an ASCII control character with a code identical to that produced by a key which controls the program, the program will respond to the code rather than accepting it as data. For instance you may use control codes to duplicate all of the codes generated by the special

keys which control the TEXT program. See Control Codes—TEXT.

You may enter control codes to the computer in BASIC in the format PRINT CHR\$(<n>) where <n> is the computer control characters. See Control Characters, Computer.

The main reason for control codes is to send commands or control codes to external devices such as a printer. By sending the right ASCII control code you can change print size, cause line feeds, and more. Normally to do this you would enter a command line from BASIC in the format LPRINT CHR\$(#) where # is the ASCII decimal representation of a control character. See Control Characters—Printer.

Characters-Per-Inch, 10 To return to normal print size on your printer, you must turn off all non-standard print options. See also Compressed Print, Double Type Formats, Type Formats.

Checksum A field of one or more bytes appended to a block of n words containing a truncated binary sum or some other function value based on the contents of that block. The sum is used to verify the integrity of data.

Chip A rectangular silicon die cut from a wafer. See entries by number for each commonly used chip.

Chip Select Every large scale integration chip normally has one (or more) chip selects, which activates one chip to receive similar signals to examine the rest of its pins (in particular the address bus, which specifies a location/register within the chip). Multiple chip-selects eliminate the need for external decoders, but require extra pins on each chip using them.

Chip-Tote* This case for your Model 100 also functions as a desk. Constructed of Cordura nylon it is padded with closed-cell Evazote foam. It features a double zipper and a non-scratch lining. The top stands erect to double as a typing stand and accessories fit into inside pockets and a detachable zippered pouch. It has both a hand strap and an adjustable shoulder strap for carrying and comes in black or grey. Kangaroo Video Products, Inc.

CHR\$ BASIC Function. Converts an ASCII code to the character it represents. The format is:

<stringvar> = CHR\$(<numex>)

<numex> must have a value in the range of 0 to 255.

For example, CHR\$(83) returns the one-character string "S." For a listing of ASCII codes refer to ASCII

Character Codes. This function is useful for accessing characters which are intercepted as control characters when keyed in, such as the double quotes in a BASIC PRINT statement. See ASCII, ASCII Character Code Tables.

CINT BASIC Statement. CINT converts a numeric expression to an integer (whole number) by truncating any digits to the right of the decimal point. The format is:

`<variable> = CINT(<numex>)`

`<numex>` must be in the range of -32768 to +32767 or an "Overflow" error occurs. See FIX and INT.

Clear Signal to place a device or a circuit in an initial, zero state.

CLEAR BASIC Command. Closes all open files and erases all memory used for data by setting all numeric variables to zero and all string variables to null. This is done without erasing the program which is currently in memory and allows you to change the default amount of string storage space in BASIC memory, and protect a segment of high memory from use by BASIC so that a Machine language program may be stored there. The format of this is:

`CLEAR [<stringspace>[,<highmem>]]`

`<stringspace>` is an optional variable that allows you to specify an amount of string storage space for use by BASIC. If you omit `<stringspace>` then BASIC retains the default stringspace of 256 bytes.

`<highmem>` is an optional variable that allows you to specify the highest memory address available to BASIC as workspace. The default value is the highest RAM memory address, 62960. This is also the value of the system constant MAXRAM. You may specify a number less than MAXRAM, 62960, to protect the memory between the two numbers from use by BASIC. You may use this space as you wish but generally high memory is reserved for storing Machine language programs or subroutines. Whatever value you use for `<himem>` will be assigned to the system variable HIMEM. HIMEM represents the highest RAM memory address available to BASIC.

The following two sample sessions illustrate how the CLEAR command works:

A. The CLEAR command erases variable contents but leaves the program intact.

```
OK
10 NUMVAR=222
20 STRINGVAR$ = WORD
30 PRINT NUMVAR, STRINGVAR$
RUN
```

```
222 WORD
OK
CLEAR
OK
PRINT NUMVAR, STRINGVAR$
0 0
OK
RUN
222 WORD
```

B. The CLEAR command lets you reserve space in high memory from use by BASIC.

```
OK
CLEAR 200,60000
OK
PRINT HIMEM, MAXRAM
60000 62960
OK
CLEAR
OK
PRINT HIMEM, MAXRAM
62960 62960
```

Clear the Screen From BASIC. See CLS.

CLK CLock. The pulsating reference timing source in a system, which triggers or synchronizes functions. Most microprocessor clocks operate in the range of 1 to 8 MHz. Real-time clocks operate at 1, 10, or 100KHz. A system usually requires a CPU clock, a timer clock, and other clocks for specific I/O devices. See Clock Frequency.

CLOAD BASIC Command. A synonym for LOAD. CAS, this erases any program lines already in BASIC memory and loads a new BASIC program from cassette tape. The format is:

`CLOAD ["<filename>"][,R]`

`<filename>` is the name of the file on cassette that you want to load to BASIC. It should be six characters or less and begin with a letter. `<filename>` is optional and if it is omitted BASIC loads the first BASIC file encountered on the cassette tape. If you do specify `<filename>`, BASIC searches the tape until it finds the file with the same name and then loads that file. When BASIC skips another file in the course of the search it will print on the LCD the name of the file skipped. This feature is useful if you can't remember what programs or files you have on a tape. Simply rewind the tape to the beginning and specify a `<filename>` that you know is not on the tape. BASIC then skips and lists all the files on the tape.

R is also optional. If included, BASIC automatically RUNs the program when it is finished loading.

CLOADM • CMP M

CLOADM BASIC Command. A synonym for LOADM "CAS:; this loads a Machine language file from cassette to high memory, beginning at the memory address which you specified when you saved the program to cassette. The format is:

CLOADM ["<filename>"]

<filename> is the name of the Machine language program you want to load to memory. It may be up to six characters in length and must start with a letter. It is optional and if you omit <filename> BASIC loads the first Machine language program encountered on the cassette tape. If you do specify <filename> BASIC lists on the LCD the names of any files it skips as it searches for the specified file name. Before loading a Machine language file, you should protect from use by BASIC a segment of RAM high memory, large enough to accommodate your Machine language program. The BASIC command CLEAR performs this function. See Address Notation, CLEAR.

CLOAD? BASIC Command. Provides a means of verifying that the BASIC program you have just loaded from cassette was loaded without error. The format is:

CLOAD?<filename>

<filename> is the name of the BASIC program currently in memory to be compared with the cassette copy it was loaded from. If the two are identical, then BASIC prints OK on the LCD. Otherwise, you will see the message, "Verify failed." If BASIC has to skip any cassette files before finding the one called <filename>, it also lists their names on the LCD.

Clock Frequency The oscillation rate of a clock, usually expressed in Megahertz. Also known as Clock Rate. The clock speed of the Model 100 CPU is 2.4 MHz.

CLOSE BASIC Statement. If used alone, this closes all open files, but if used with <filenum>, only those files specified are closed.

CLOSE[#<filenum>[,#<filenum>,...]]

<filenum> represents a buffer used for data input and output to a specified file or device. Both <filenum> and its associated device or file are defined by the same OPEN statement.

When CLOSE is executed, the file or device and its file number, i.e., the buffer number used for the data transfer, are no longer paired. Later I/O operations like file, buffer, number will not be valid. The file or device can then be re-opened using the same or a different <filenum>. Alternatively, the same file (buffer) number may be reused to open a

different device or file. Closing a file or device, which was opened for sequential output as <filenum>, will write the buffered data to the file or device.

The maximum number of files allowed in BASIC, represented by the system variable MAXFILES, is 15. The default value is 1.

A CLOSE with no file numbers specified causes all devices and files that have been opened to be closed.

Executing an END, NEW, or RUN causes all open files and devices to be automatically closed. STOP does not close any files or devices.

CLOSE is used in conjunction with the OPEN statement. See OPEN and File I/O Commands and Functions.

Close Disassociation of a file from a particular program, including flushing unwritten buffers and updating the directory and File Allocation Table if required.

CLS BASIC Statement. Used to clear or "blank" the LCD screen, this turns off all pixels and returns the cursor to the "home position," the upper right hand corner of the screen. The format is:

CLS

CMA 80C85 Assembly Language Instruction. CoMplement Accumulator. The contents of the accumulator are complemented (0 bits become 1, 1 bits become 0). No flags are set.

CMC 80C85 Assembly Language Instruction. CoMplement Carry. The CY flag is complemented. No other flags are set.

CMOS Complementary MOS. Technology characterized by extremely low power consumption—widely used in portable applications and for battery-assisted memory systems. CMOS requires both p-channel and n-channel transistors, with speed and density characteristics intermediate between NMOS and PMOS. CMOS devices operate from 3v to 12v and have ideal noise immunity characteristics.

The CPU, ROM, and RAM in the Model 100 all use CMOS technology to reduce the power requirements of the computer, and increase battery life. The LCD and piezoelectric speaker also represent a power savings over the traditional cathode ray terminal and permanent-magnet speaker used in most computers. The Model 100 operates a GVDC.

CMP M 80C85 Assembly Language Instruction. CoMPare Memory. The content of the memory location whose address is contained in the H and L

registers is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if the number contained in the accumulator is equal to the number contained in the memory location specified by the H and L registers. The CY flag is set to 1 if the number contained in the accumulator is less than the number contained in the memory location specified by the H and L registers. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

CMP R 80C85 Assembly Language Instruction. CoMPare Register. The content of the register <r> is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if the number in the accumulator equals the number in <r>. The CY flag is set to 1 if the number in the accumulator is less than the number in <r>. The addressing mode is register. Z, S, P, CY, and AC flags are set.

CMR Common Mode Rejection. The devices' ability to handle multi-input terminal voltage.

CMRR Common Mode Rejection Ratio. Operational amplifier's common mode gain.

CMRR Common Mode Rejection Ratio. Operational amplifier's common mode gain.

CO File Extension. Designates the file as a Machine language file. The .CO extension is automatically assigned by the operating system when you use the LOADM command, to load a Machine language program from cassette to a RAM file.

Coax Coaxial cable. A transmission line with an inner conductor and an outer shield conductor.

COBOL COmmon Business Oriented Language. A high-level programming language developed for business applications with statements based on English.

Code Synonym for Program Instructions Code. This applies to program instructions, language statements, or a symbolic representation for data, in any language, including BASIC, Machine language, and others. See Program.

Code Key See ASCII and ASCII Character Code Table.

CODE Key The CODE key is located in the lower right hand segment of the Model 100 keyboard, directly to the right of the space bar. By pressing the CODE key and a standard keyboard key, or the CODE key, a standard keyboard key, and the SHIFT

key at the the same time, you can generate sixty-six special graphics characters. These are 8-bit ASCII codes 160-162, 164-175, and 181-223. They will all display on the LCD. See ASCII Character Code Table.

Codec COder-DECoder. A chip providing the essential translation of analog to digital conversion.

Cold Start Starting the computer after all power to RAM memory has been shut off and its entire contents erased. Press the BREAK/PAUSE and the CTRL keys together with one hand, and turn the power ON/OFF switch ON with the other. You may achieve the same result by turning the memory power switch OFF and then ON again. This not only erases all RAM files, but also the system values for the TIME\$, DATE\$, and DAY\$ variables. See Memory Power Switch.

Colon (:) BASIC Device Name Delimiter. In a filename, the : and . are specified to delimit device name/file name/and file extension respectively. For example:

SAVE "RAM:exam.do"

Here, the device is RAM, delimited by a colon (:). The file is exam, with extension do, delimited by a period (.). See Device Names.

Colon (:) Used to put multiple statements on one line. You can put a colon (:) at the end of one statement and continue entering another statement on the same line, without giving a new line number. The line number at left refers to all statements on the line. If a BASIC line exceeds forty characters it automatically wraps to the next line on the LCD and continues to do so until terminated and entered by the ENTER key. An apostrophe (') allows a comment or remark to be added after a statement with or without a separating colon. See BASIC Statements.

COM—BASIC Statement. The COM statement is used to enable to disable trapping of the RS-232-C interrupt. When the interrupt is enabled by the COM statement, communications activity via the RS-232-C port will cause program execution to branch to the subroutine specified by a previously executed ON COM statement. The formats are:

COM ON
COM OFF
COM STOP

Trapping of communications activity by the ON COM statement does not occur until COM ON is executed. Trapping continues until the program ends or a COM OFF or COM STOP is executed.

COM: • Command Program

COM OFF totally ignores all RS-232-C activity. COM STOP notes communications activity but holds any trap until a COM ON is executed, at which time the trap occurs immediately with program execution going to the ON COM subroutine. See RS-232C.

COM: Reserved Device Name. Stands for the RS-232C communications port. COM may be the device specified in an I/O transmission filename. Other devices that may be used for transmitting file I/O are MDM, the modem; LPT, the printer; and LCD, the display screen. See File I/O Commands and Functions.

Combinational Logic A circuit with Boolean logic functions and no memory.

Comma (,) Used to space print-outs. Each group of fourteen spaces across the print line is called a print zone. A comma (,) in an LPRINT list of items to be printed means "start printing the following item at the start of the next print zone." Contrast this with the semicolon (;), which means the next item is to print immediately after this one, without a space between. In reality, numeric variables are bracketed by one space on each side. The print zones begin in columns 0, 14, 28, 42, 56, 70, etc. See also Print Lines, Print Zones.

Comma, Double (,,) Used in print line to leave a lot of space on the print line between items. For example:

LPRINT A, ,B

Prints A in print zone 1 (beginning at column 0), nothing in print zone 2, and B in print zone 3 (col 28). See also Print Zones—On Printer Line and PRINT Line—Spacing and SPACE\$.

Command A statement that causes a computer to carry out a specific action. Commands differ from instructions in several ways. A command is usually a complete specification of an action, while dozens or hundreds of instructions are combined to make a useful program. Commands are usually acted upon immediately by the computer, while instructions are saved for later execution in a program. Commands are acted on by the operating system of the computer, while instructions must be processed by a particular program such as the BASIC interpreter, a FORTRAN or PASCAL compiler, etc. See BASIC Command Table.

Command Mode BASIC The BASIC that comes with the Model 100 is an Extended BASIC created by Microsoft. BASIC has three modes of operation: Command mode, Execute mode, and Edit mode.

When you first enter BASIC from the Main Menu you will be in Command mode.

Command mode displays the prompt OK to indicate that you may enter BASIC commands or BASIC program lines. Program lines always begin with a program line number. Program lines may be numbered between 0 and 65529. When you key in a program line and enter it to BASIC it is stored in BASIC memory and not executed or acted upon by the computer until you enter the BASIC RUN command. When you RUN or execute the program lines they will execute in line number order from first to last line, unless you include BASIC statements to branch execution to nonsequential program lines.

To list the program lines currently stored in BASIC memory on the LCD, key and enter the BASIC command LIST. To list program lines on an attached printer, key and enter LLIST. BASIC commands may be entered without program lines while in Command mode. In this case when you enter the command line to BASIC it is immediately executed. The computer immediately performs the actions represented by the keywords you have entered. Commands entered without line numbers are not saved to BASIC memory and will not list.

Command Processor A program which accepts a command (usually from a keyboard) and causes it to be carried out. Some command processors contain the programming required for all commands they process. Others do not carry out any commands directly. Instead, they examine the command, determine what other program (if any) can carry out the command, locate the required program, and start it running. Still another type of command processor carries out some commands directly (internal commands), but locates and runs other programs for some commands (external commands). See BASIC Commands.

Command Program All programs that run directly on a computer are Machine language programs, in the actual numeric instruction code of the 80C85 microcomputer chip. Most were originally written by a programmer as text files known as source programs. The source program contains relatively readable statements in a language such as FORTRAN, COBOL, or BASIC. These were then translated by a compiler program which produced an object program.

The object program contains the Machine language instructions for the Model 100's 80C85 which corresponds to the instructions of the original source program. BASIC programs work in this way with a BASIC compiler.

Comment Field • Communications Parameters, How to Set

Regular BASIC works in a slightly different way. Regular BASIC is a program (in Machine language and stored in ROM) which uses your BASIC program as a guide to what it should do. It is therefore an interpreter, processing each line of your source program and interpreting what should be done. Since it must re-interpret your source program each time you run it, interpreted BASIC can be as much as 100 times slower than compiled BASIC.

When you write a BASIC program, you have produced a source program in text form. The BASIC interpreter, itself a Machine language program, uses your source program as data—a source of instructions or commands to control its execution. Suppose you decide to speed up a BASIC source program by compiling it into a faster Machine language program. You will need a BASIC compiler, a program not supplied with the Model 100, but available from various software houses. You will run your BASIC compiler, giving it the source program (.BAS) that you wrote as input. The BASIC compiler will translate your program.

You will then use the LINK program to further process the .OBJ file into an executable .EXE runfile. The resulting .EXE program is no longer a BASIC program—it is a Machine language program which is based on (or is a translation of) your BASIC source program.

Comment Field A field within an instruction, which contains explanations or remarks, that is ignored by the interpreter, the compiler, or the Assembler.

Comments in BASIC Programs Use REM to include remarks or explanatory comments in a program. The apostrophe (') is a synonym for REM and can be used interchangeably with it. The apostrophe can be used without a colon to indicate that the rest of the line is a remark. For example:

```
20 REM Just a comment
30 ' Just a comment
40 LET X = 1: REM Just a comment
50 LET X = 1: ' Just a comment
60 LET X = 1' Just a comment
```

The colon (:) allows multiple BASIC statements on one line.

Communications Parameters, How to Set Any time you transmit data between computers via the built-in modem or the RS-232C interface, you must be sure that both computers are using the same communications parameters. Computer-to-computer data transfers may be initiated from within TELCOM Entry mode or from within BASIC using

the LOAD, SAVE, RUN, MERGE, and OPEN commands.

The communications parameters are a series of one-character values represented by the variable format <r> <w> <p> <s>. Parameters and their values are:

<r> is the baud rate. Values range between 1 and 9 where each integer number stands for a different baud rate.

M = modem (300 baud)

1 = 75 baud

2 = 110 baud

3 = 300 baud

4 = 600 baud

5 = 1200 baud

6 = 2400 baud

7 = 4800 baud

8 = 9600 baud

9 = 19200 baud

If you plan to enable the modem as the communications device using BASIC, omit the <r> value. The modem will operate at 300 baud by default. If you plan to enable the modem as the communications device from within TELCOM, specify M as the <r> value. If you specify any other <r> value the modem will be disabled and the RS-232C baud rate will be set at the value you selected.

<w> represents the word length in bits and may be set to values 6, 7, or 8.

<p> represents parity and may be set to:

E = even parity

O = odd parity

I = ignore parity

N = no parity

 indicates stop bits. 1 for one stop bit and 2 for two stop bits.

<s> enables and disables XON/XOFF status. Valid values are E which enables it and D which disables it.

To set the communications parameters from within TELCOM Entry mode, press F3 (Stat key). This displays the current default communications parameters (M711E,10 pps). The segment of the string before the comma contains the values corresponding to <r> <w> <p> <s>. The segment of the string after the comma represents the pulses per second that the modem will use to automatically dial the phone. You may change this value to 20 pps if your local phone system supports this. If you want to change any of the communications parameters displayed by pressing the Stat key (F3), you must backspace over and rekey the entire string. It is important to realize that TELCOM dis-

tingishes between upper and lower case letters, so use exactly the format given. It is not necessary to rekey the pulses per second specification if you are satisfied with the default value.

To set the communications parameters from within BASIC, just follow the format given in the individual BASIC keyword entries (LOAD, SAVE, OPEN, and MERGE) elsewhere in this encyclopedia. In each case the BASIC command will include the communications parameters in the <r> <w> <p> <s> format and the same valid values will apply (with the exception of M for the <r> option).

Communications Software The Model 100 has built-in communications software. See TELECOM. See also Traveling Communicator, The, TELEX+.

Compatible, Upward The term upward compatible indicates that programs developed for one version of a programming language, operating system, application software package, or computer will work, without alterations, on an expanded, more powerful version of the same language, system, or package.

In hardware, upward compatibility refers to the possibility of building up to more powerful models without reprogramming.

Compile Time The point in the processing of a program when it is being translated from source code to object code by a translator (compiler or Assembler).

Compiler A translation program which converts high-level instructions into binary instructions for direct processor execution. Any high-level language, such as BASIC or COBOL, requires a compiler or an interpreter.

An interpreter translates each statement of the program each time the statement is executed, whereas a compiler translates the complete program at once, producing object code that can be executed repeatedly. Any change in the program requires recompilation. The code produced by compilers may be longer and/or slower than the code generated from Assembly language source code.

Compiler Programs A program which translates a source program written in a higher level language, such as BASIC or FORTRAN, into Machine language.

Complementing The action of changing each 1 to a 0, and each 0 to a 1.

Compressed Print To get this small type size on your printer, enter the BASIC statement:

LPRINT CHR\$(143)

either in a program, with a line number, at the point you want to start printing in the small type, or in direct mode without a line number. This gives 132 characters on the 8-inch line, or about sixteen characters-per-inch. To return to normal size print, enter LPRINT CHR\$(146). See Type Formats.

Compressed Print—72/7 lines-per-inch This is a good setting for spacing with compressed print.

Use the BASIC statement:

#LPRINT CHR\$(155);"1"

or

#LPRINT CHR\$(155);CHR\$(49)

See Type Formats.

COMPRINT 912-GO* This graphics printer features 100 dots-per-inch resolution (horizontally and vertically). Comprint.

COMPRINT 912-Serial* The COMPRINT 912 is a serial electro-graphics computer printer including a 12 by 9 matrix, 225 characters-per-second, 170 lines-per-minute, 80 characters-per-line, RS-232 and 20ma current loop serial interface, and 256 byte buffer. Comprint.

CompuServe* A subscription information service which can be of great value to microcomputer users. You can access the service with your Model 100 using the built in modem or an acoustic coupler and modem cable. CompuServe requires payment of a \$20.00 initial subscription fee that entitles you to a special ID number and password. The on-line cost of CompuServe runs from \$5.00 an hour during non-peak times such as nights, weekends, and holidays, to \$22.50 an hour during peak business hours. Compared to similar services, the startup and hourly fees of CompuServe are low and you are not charged a monthly minimum.

A wide selection of services are offered, including electronic catalog shopping, education, games, entertainment, electronic bulletin board, Dow Jones updates, electronic mail, and word processing. Other more specialized services are also available, including: a simulated CB radio network; systems operators (SYSOPs) available to help with questions; and special interest groups (SIGs), with their own assistants.

For additional fees the service can be customized to include teleconferencing, business and financial information highlights, personal finance programs, travel arrangements, news, and use of supplemental networks such as Tymnet and Datapac.

Instructions for various CompuServe operations follow:

AUTOMATIC LOGON. An example of the command sequence used to perform the automatic log-on function is given below. You must include the correct telephone number for your area, your identification number, and password, as indicated within the brackets []. (Do not include the brackets.) Square brackets are used here because the source uses the arrow brackets.

First, enter these lines into the ADRS.DO file using the TEXT application found in the Owner's Manual. After you enter the lines, you dial using the TELCOM application. Locate the correct line using FIND and then use the CALL function key to begin. For TELCOM, the suggested STATUS setting is M711D,10.

```
CompuServe
CSIS: [CompuServe phone]≤AC?U[user id]
      AM?P[password]AM>:
CompuServe via Tymnet (where direct service
is not available)
CSTYM: [Tymnet phone]≤=
      A?pCPS?pWELCOMEAM?U[user id]
      AM?P[password]AM>:
```

At the end of the sequence enclosed in the ≠ characters, you may add additional requests.

MODEL 100 SIG (SPECIAL INTEREST GROUP). There are over 50 different special interest groups (SIGS) offered by CompuServe with one designated specifically for Model 100 owners. After logging on to CompuServe and seeing the command prompt, enter:

GO PCS 154

this takes you to the Model 100 SIG, and a function menu is displayed:

MODEL 100 FORUM FUNCTION MENU

- 1 (L) Leave a message
- 2 (R) Read messages
- 3 (RN) Read new messages
- 4 (RM) Read waiting messages
- 5 (B) Read bulletins
- 6 (CO) Online conference
- 8 (MI) How to join this SIG
- 9 (OP) Change your SIG options
- 0 (E) Exit from SIG

Enter selection or H for help:

By selecting nine on the Function Menu you will call up the User Options Menu:

MODEL 100 USER OPTIONS MENU

- 1 Change to command mode
 - 2 (LL) Change line length
 - 3 (T) Return to function menu
 - 0 (P) Make options permanent
- Enter selection or H for help:

(LL) gives current line length default and allows you to choose a new one. Enter 40 and you will be ready to go.

LEAVE A MESSAGE ON THE BULLETIN BOARD. In most cases, it is a good idea to create your message before going on line and posting it to the message board or UPLOADING. Write the message with TEXT and be sure to create a separate file for each. Once online you select the Leave Message function from the Forum Function Menu, decide if you want to place a general message or ask assistance from a particular SIG user (entering their user ID number), and answer the systems question about the subject of your inquiry. The next system response is:

```
Enter your message.
Use a blank line or control-Z to end message.
1:
```

F3 (UPLOAD) chooses which message you want to send and enter the file's line width. At the end of the transmission, the Model 100 automatically exits the upload mode and displays the Leave Option Menu:

LEAVE OPTIONS:

- 1 (S) Store the message
- 2 (L) List the message
- 3 (R) Replace a line
- 4 (D) Delete a line
- 5 (C) Continue entering text
- 6 (A) Abort the leave function

Enter selection or H for help:

COPY A PROGRAM FROM THE MODEL 100 SIG DATABASE. DOWNLOADING copies programs from the SIG databases to your Model 100. There are three steps to this procedure: reviewing what's in the files, copying it either into the memory of your computer or directly to a printer for hard copy, and turning the text into a BASIC program.

First, enter XA at the function prompt. When the SIG/Access prompt is given, specify which database you want to access. The best way to get a complete abstract of the files is to catalog the files by keywords or descriptions. For example, enter:

TYP CAT/KEY:GAME/DES

to review a scrolling list of all the games available with a description of each. To download the program you have chosen (let's call it DECWARS.100) into the memory of the Model 100, enter:

TYP DECWARS.100 <F2> (system will ask for name of program again)

DECWARS.100 ENTER (CompuServe is waiting for enter after TYP DECWARS.100D)

Computer • Continue a Program after a Pause or Break

ENTER <F2> (enter this after SIG/Access prompt to exit download)

Note: <F2> means the function key F2.

To turn the text file into a BASIC program you must move the cursor over DECWARS.DO and open the file with ENTER. Check for inappropriate line feeds and mark all the program lines as a text block using F7. Cut the text block into the paste buffer using F6 and press F8 to exit DECWARS.DO and go to BASIC. Now type NEW to erase the current BASIC memory contents, then hit the paste key and you will receive the programming lines from DECWARS.DO. To save the program hit F3 (SAVE) and enter file name. This saves the program to a RAM file with a .BA extension.

The actual documentation you receive with CompuServe is somewhat below average, but help menus are available and for assistance you can call a toll-free telephone number (800) 848-8990. Requires 300 Baud. CompuServe.

Computer A general purpose computing system incorporating a central processing unit, memory, I/O facilities, power supply, and cabinet.

Computer Memory Storage in Bytes. Byte is a label for storage to hold one character (letter, digit, etc) in computer memory, internal or diskette. Abbreviated B or in thousands KB, or simply K. Actually, 1K = 1024 because this is an even power of 2. The Model 100 is currently available with 8, 16, or 32K of internal RAM memory. See Address.

Computer System A complete system including the MPU, keyboard, CRT, and other peripherals such as printers, disks, tapes, etc. Often used to also include programs.

Concatenating Strings In BASIC you may concatenate combine string variable contents to form a new variable. Simply "add" the string variables using plus signs and assign the concatenated result to another variable name. The second string will be appended to the end of the first.

```
Ok
A$="I think "
B$="therefore I am"
C$=A$+B$
PRINT C$ I think therefore I am
Ok
```

Concatenation Means adding one item (data or programs) to others to produce a longer item.

Console The terminal that has the most control in a system. For a microcomputer, the keyboard or the front panel is the console.

Constant An explicit value in a program instruction or statement rather than a symbolic value. The value is fixed throughout a program.

CONT BASIC Statement. Resumes program execution after a break caused by pressing the SHIFT and BREAK/PAUSE keys together or execution of a STOP or END statement. A break from the keyboard or one caused by a STOP statement in the program will display the message "BREAK in <xxxx>" where <xxxx> is the number of the last line executed. A break caused by the END statement prints no message.

When CONT is entered following a break in program execution the program will continue executing at the point where the break occurred. If the break occurred following a prompt from an input statement, execution will continue following the reprinting of the prompt.

CONT is useful in conjunction with STOP for debugging purposes in a process known as break-pointing. The programmer may insert STOP statements throughout the program to temporarily halt program execution at significant intervals. When execution is stopped you can examine variable contents using direct mode PRINT statements and even change variable values with direct mode assignment statements. You may then use CONT or a direct mode GOTO to resume program execution at the place the break occurred or at another particular line number.

CONT will not work to resume program execution if any program lines have been changed during the break.

Continue a Program after a Pause or Break The BREAK/PAUSE key temporarily pauses current BASIC program execution when pressed once and resumes it from the current line when pressed a second time. This is useful if, for instance, you want to pause while listing a BASIC program so that you can read it before it scrolls off the screen. Pressing SHIFT and BREAK/PAUSE at the same time breaks program execution, as does executing STOP or END as a program line. BASIC programs will display the message "BREAK in <xxxx>" where <xxxx> is the program line that was executing at the time of the BREAK, unless you used the END option to pause execution. Enter the BASIC statement:

CONT

to resume program execution at the line where the interrupt occurred. You may also resume program execution by entering:

GOTO <linnum>

where <linnum> is the program line number

Continue Same Logical Line ... • Converting from Double- to Single-Precision ...

where you want to resume execution. Neither CONT nor GOTO will resume program execution if you have altered program lines during the break, although you may use direct mode print or assignment statements to examine or change variable contents.

Continue Same Logical Line On The Next Screen Line When entering a BASIC program line or direct mode command and the number of characters exceeds forty without a carriage return, i.e., you have not yet entered it to BASIC, the line automatically wraps to the next LCD line. As long as you don't press ENTER, it is still regarded as one logical line.

Control Bus The set of control lines (in a computer system) to carry the necessary synchronization and control information throughout the system. Examples of signals carried on these lines are orders such as Read or Write, and sync signals such as Interrupt, Hold, or Acknowledge.

Control Characters—Computer Computer characters having specific system-dependent meanings. Some of the control characters in the Model 100 are:

ASCII code (dec)	Meaning
7	bell
8	backspace (destructive)
9	tab
10	linefeed
11	home cursor
12	clear screen
13	carriage return
27	escape

To enter control characters to the printer use the CHR\$ and LPRINT BASIC keywords. For example, entering:

```
PRINT CHR$ (7)
```

causes the computer to BEEP.

Control Characters—Printer To set the Epson RX-80 printer's print size, strike method, or number of lines per inch, the non-standard type format you want must be turned on by sending control codes to the printer. This is done in BASIC mode by entering a BASIC CPRINT line containing ASCII control codes, and other special codes. See table next page.

Control Codes—TEXT Most of the functions normally performed in TEXT by the cursor and programmable function keys can also be performed by a combination of the CTRL and alphabet keys. The following table shows TEXT control codes, their function, cursor and other special key equivalents and the function performed. See table next page.

Control Keys Keyboard. Most of the control keys are defined by the currently executing program, rather than by the Model 100's hardware. The same key may produce one function in TEXT, another in BASIC, and still another in a program. See Keyboard.

Control Unit The module which fetches and decodes instructions. The CU requires an instruction register and a program counter. It generates control signals and manages the control bus.

Controlled Test Environment The software which is in the computer with a program to be tested, which may determine your ability to detect and interpret errors.

Controller Circuit board or boards that join a peripheral to the computer, having complex circuitry to maintain device control.

Convert Change from one form to another. The BASIC Edit Command converts a tokenized BASIC file into an ASCII file. Convert also applies to hardware—it is possible to convert to a graphics printer by installing the GRAFTRAX option.

Convert Number to Character BASIC. See CHR\$.

Converting from Double-Precision to Integer Values in BASIC This conversion is done by assigning a double-precision value, variable or constant, to an integer variable. BASIC will remove all digits to the right of the decimal point. The resulting number must be in the valid range for integer values, anywhere from -32768 to 32768. If the value is out of range you will get the message, "?OV Error," the overflow error message. A sample conversion is:

```
OK
d%=34.984532871234
OK
Print d%
34
OK
```

Converting from Double-to Single-Precision Numbers in BASIC This conversion may be easily accomplished in BASIC by assigning a double-precision number, constant or variable, to a single-precision variable. BASIC will round the digits to the right of the decimal so the number is reduced from four-

Converting from Integer to Double-Precision Numbers in BASIC

teen to six significant digits. It uses a rounding method known as 4/5 rounding which rounds down numbers from one to four and rounds up numbers from five to nine. For example:

```
OK
n!=56.467853987637
OK
Print n!
56.4679
OK
```

Converting from Integer to Double-Precision Numbers in BASIC This conversion is done by assigning an integer value, variable or constant, to a double-precision variable. Remember that the default variable type in BASIC is double-precision

numeric. BASIC adds the number of zero digits to the right of a decimal point necessary to equal fourteen significant digits in all. For example:

```
OK
d#=34
OK
Print d#
34.00000000000000
OK
```

When performing mathematical operations using variable or constant values of different numeric types, BASIC temporarily converts all numbers to double-precision for the duration of the calculation. They are not changed in the original memory locations, and when stored back into variables take on the type indicated by the variable.

Control Characters—Printer

Print Type	Characters Per Inch	To Turn On (Enter BASIC Line)	To Turn Off (Enter BASIC Line)
Pica	10	LPRINT CHR\$(27);"P";	Default value
Elite	12	LPRINT CHR\$(27);"M";	LPRINT CHR\$(27);"P";
Enlarged	5	LPRINT CHR\$(27);CHR\$(14);	LPRINT CHR\$(20);
Condensed	17	LPRINT CHR\$(27);CHR\$(15);	LPRINT CHR\$(18);
Print Type	To Turn On (Enter BASIC Line)	To Turn Off (Enter BASIC Line)	
Emphasized	LPRINT CHR\$(27);"E";	LPRINT CHR\$(27);"F";	
Double Strike	LPRINT CHR\$(27);"G";	LPRINT CHR\$(27);"H";	
Italics	LPRINT CHR\$(27);"4";	LPRINT CHR\$(27);"5";	
Underline	LPRINT CHR\$(27);"-"; CHR\$(<i><n></i>) (<i><n></i> is 1 or 49)	LPRINT CHR\$(27);"-"; CHR\$(<i><n></i>) (<i><n></i> is 0 or 48)	
Line Spacing	To Turn On (Enter BASIC Line)	To Turn Off (Enter BASIC Line)	
1/6 Inch	LPRINT CHR\$(27);"2";	Default Value	
1/8 Inch	LPRINT CHR\$(27);"0";	LPRINT CHR\$(27);"2";	
7/72 Inch	LPRINT CHR\$(27);"1";	LPRINT CHR\$(27);"2";	
<i><n></i> /72 Inch	LPRINT CHR\$(27);"A"; CHR\$(<i><n></i>) <i><n></i> ranges from 0 to 85	LPRINT CHR\$(27);"2";	
<i><n></i> /216 Inch	LPRINT CHR\$(27);"3"; CHR\$(<i><n></i>) <i><n></i> ranges from 0 to 255	LPRINT CHR\$(27);"2";	

See Type Formats for a full explanation and examples of type formats and line spacings.

Converting from Integer to Double-Precision Numbers in BASIC

Control Codes—TEXT

ASCII Code	CONTROL Key Combination	TEXT Key(s) Equivalent	TEXT Function Performed
1	CTRL A	← SHIFT	Moves cursor to start of first word to left.
2	CTRL B	← SHIFT	Moves cursor to last LCD line or last line of next screen if there already.
3	CTRL C	SHIFT BREAK/PAUSE	Cancels FIND, SAVE, LOAD, PRINT or SELECT function.
4	CTRL D	→	moves cursor right one character
5	CTRL E	↑	moves cursor up one line on LCD.
6	CTRL F	← SHIFT	moves cursor to start of first word to right.
7	CTRL G	F3	saves current file
8	CTRL H	DEL BKSP	deletes character to left of cursor
9	CTRL I	TAB	moves cursor to the next TAB stop word to right
10	CTRL J	-----	not used
11	CTRL K	-----	not used
12	CTRL L	F7	mark the beginning of a text block
13	CTRL M	RETURN	end a paragraph and carriage return
14	CTRL N	F1	find a string [if after CTRL R stop underline printer control]
15	CTRL O	F5	copy text block to paste buffer [if after CTRL P start underline printer ctrl]
16	CTRL P	-----	preface printer control character
17	CTRL Q	CTRL ←	moves cursor to far left of current line
18	CTRL R	CTRL →	moves cursor to far right of current line
19	CTRL S	←	moves cursor left one char
20	CTRL T	↑ SHIFT	moves cursor to top LCD line or top of the next screen if there
21	CTRL U	F6	delete a block of text, copy it to paste buffer
22	CTRL V	F2	load a cassette file to TEXT
23	CTRL W	CTRL ↑	moves cursor to first file character
24	CTRL X	↓	moves cursor down one line on LCD
25	CTRL Y	PRINT SHIFT	prints the entire file on printer
26	CTRL Z	CTRL ↓	moves cursor to the last file character

Converting from Integer to Single-Precision Numbers in BASIC • Crash

Converting from Integer to Single-Precision Numbers in BASIC This conversion may be easily accomplished by assigning an integer value, variable or constant, to a single-precision variable. BASIC adds the number of zero digits. For example:

```
OK
d!=34
OK
Print d!
34.0000
OK
```

Converting from Single-Precision to Integer Values in BASIC This conversion may be easily accomplished by assigning a single precision value, variable or constant, to an integer variable. BASIC will truncate all digits to the right of the decimal point. The resulting number must be in the valid range for integer values, anywhere from -32768 to 32768. If the value is out of range you will get the message, "?OV Error," the overflow error message. A sample conversion is:

```
OK
d%=34.1234
OK
Print d%
34
OK
```

Converting from Single to Double-Precision Numbers in BASIC This conversion may easily be accomplished by assigning a single-precision number, constant or variable, to a double-precision variable. Remember that the default variable type in BASIC is double-precision numeric. BASIC will add zeros to the right of the decimal point until the number is increased from six to fourteen significant digits. For example:

```
OK
d#=36.8907
OK
Print d#
36.890700000000
OK
```

COPY* Designed to increase merging and duplication capabilities. Can be used with a modem. Cassette. Micro Computer Services.

Core A small, magnetic torus (or doughnut) of ferrite used to store a bit of information. Cores can be strung on wires so that memory organizations of 32K 18-bit words can be packed into space with dimensions ½ by 6 by 6. One advantage of core is that it is non-volatile. Also, as a holdover from the days when nearly all computer internal memory

was core, the word is still sometimes used as a synonym for internal memory: core-image, in core, etc.

COS BASIC Function. Returns the trigonometric cosine function of <numex>. The format is:

<variable> = COS(<numex>)

<numex> is the angle whose cosine is to be calculated. The value of <numex> must be in radians. To convert degrees to radians, multiply the degrees by pi/180, where pi = 3.141593.

Counter Binary counter.

CPI 80C85 Assembly Language Instruction. Compare Immediate. The content of the second byte of the instruction is subtracted from the accumulator. The condition flags are set by the result of the subtraction. The Z flag is set to 1 if the number contained in the accumulator equals byte 2. The CY flag is set to 1 if the number in the accumulator is less than byte 2. The addressing mode is immediate. Z, S, P, CY, and AC flags are set.

CP/M Control Program for Microcomputers. A popular single-user operating system for 8080, Z80, and 8085-based microcomputers created by Digital Research. CP/M can be run on the TRS-80 via CP/M-86 or through an auxiliary processor board which supports one of the original CP/M processors (Xedex, Big Blue, etc.). A third option is to run a program which emulates one of the original CP/M processors. There is a huge base of software for CP/M, making it very attractive as a main or supplementary operating system.

CPS Characters Per Second or Cycles Per Second.

CPU Central Processing Unit. The device in charge of fetching, decoding, and executing instructions. It contains a control unit, an ALU, and other related facilities with registers, clocks, or drivers. The Model 100 uses the 80C85 CPU chip, a CMOS version of the 8085 CPU.

CPU I/O BASIC Command Table

CPU I/O	Function Performed
OUT	Outputs a data byte to the central processing unit
INP	Inputs a data byte from the central processing unit

CR Carriage Return. 13 decimal or 0D base 16, or Command Register, or Card Reader.

Crash A situation where the system becomes inoperative through hardware or software malfunction. A head crash refers to the accidental

impact of the read/write head upon the diskette surface.

CRC Cyclic Redundancy Check. A binary polynomial used to check information in blocks of data. All single-bit errors are detected. When used with an additional Longitudinal Redundancy Check (LRC), all two-bit errors are detected and all one-bit errors corrected. The two check sums (CRC and LRC) are calculated based on the data and appended to it. When the data is received or reread, the CRC and LRC are recalculated and compared to the earlier CRC and LRC. Any difference indicates that a bit has changed. The CRC and LRC act like parity bits, except that they work on a whole block of data rather than one byte.

Create Making a new file. This can be done in TEXT by specifying a new file name in response to the prompt "File to Edit?" that you see when you first enter text.

Creating and Testing BASIC Error-Handling Routines See ERROR, ON ERROR, GO SUB, ERR, ERL.

Creative Computing Computer Magazine. There are a number of popular magazines which contain useful information concerning Model 100 and available hardware and software for it. Most computer stores and larger bookstores carry a good assortment of these magazines. *Creative Computing's* address is:

Creative Computing
P.O.Box 789-M
Morristown, NJ 07960

U.S.: 1 year \$20 for 12 issues.

See Magazines for addresses and subscription rates of other magazines which carry information about the Model 100.

CROM Control Read-Only Memory.

Crosstalk Two signals interfering with one another.

CRT Cathode Ray Tube. A computer terminal using a CRT to display characters or pictures. Also called a monitor, screen, terminal, etc.

CRTC CRT Controller. A chip or circuit which provides the necessary control signals to interface a CRT to the I/O bus of an MPV.

Crystal Quartz crystal whose piezoelectric vibrational modes issue extremely accurate frequencies for clock timing.

CS Chip Select or Code Segment.

CSAVE BASIC Command Synonymous with BASIC command SAVE "CAS: this SAVES the contents of

BASIC memory to cassette tape. The format is:

CSAVE "<filename>" [,A]

<filename> is the name that will be assigned to the file saved on cassette from BASIC memory. It should be six characters or less and begin with a letter. If you omit the ,A option BASIC saves the file to cassette in a tokenized BASIC format that takes less cassette tape to store.

,A is an optional specification. If you do specify the ,A option the current contents of BASIC memory are saved to cassette tape in the ASCII format used by BASIC commands such as MERGE.

CSAVEM BASIC Command. CSAVEM works exactly like BASIC command SAVEM "CAS:". It saves a Machine language program, located between a given start and end address in memory, to cassette tape. The format is:

CSAVEM "<filename>, <startadd>,
<endadd> [, <entryadd>]

<filename> is the name that will be assigned to the file saved to cassette from the high RAM memory address range defined by <startadd> and <endadd>. It should be six characters long or less and begin with a letter.

<startadd> is the start address in memory that holds the first byte of data you will load to cassette.

<endadd> is the end address in memory that holds the last byte of data you want to load to cassette.

<entryadd> is an optional specification. If included it represents an entry address into the Machine language program other than the <startadd>. See Address Notation.

CSNG BASIC Function. Converts a numeric expression to a single-precision number. The format is:

<variable> = CSNG(<numex>)

<numex> is the numeric expression which is to be converted to single-precision. BASIC converts integers to single-precision values by adding the number of zero digits to the right of a decimal point necessary to equal six significant digits. BASIC converts double- to single-precision values using 4/5 rounding to reduce fourteen significant digits to six.

CSRLIN BASIC Variable. Returns the current line (row) position of the cursor on the LCD. The format is:

<variable> = CSRLIN

The value returned is in the range 0 to seven where 0 represents the top LCD line. The related function is the POS function, which returns the column location of the cursor. See POS.

CTRL Key • Cursor—Move Up One Space

CTRL Key Pressing the CTRL key with an alpha key produces a control code with an ASCII value between 1 and 26. Control codes may also be entered using BASIC PRINT or LPRINT CHR\$(<n>) statements where <n> is the ASCII code decimal equivalent. These codes do not display on the LCD or the printer. Control codes may be interpreted differently when used in different applications. In TEXT mode, the control codes produced provide an alternate means of keying the codes usually generated by the function keys, cursor keys, and other special keys such as RETURN.

Most of the functions normally performed in TEXT by the cursor and programmable function keys can also be performed by a combination of the CTRL and alphabet keys. The following table shows TEXT control codes, their function, cursor and other special key equivalents and the function performed. See table next page.

Control codes are also used to change printer configurations, such as lines per page and character size. In this case, however, you would send the codes to the printer from BASIC using the LPRINT CHR\$(<num>) configuration, where <num> represents the ASCII value of one of the control codes. You could also key LPRINT and a CTRL alpha key combination, but this would be executed immediately and so cannot be built into a program for execution when the program is RUN. See Control Characters—Printer.

To enter control codes to the computer use BASIC statement PRINT CHR\$(<n>) where <n> is the ASCII decimal numeric equivalent. See Control Characters Computer.

CTS Clear To Send. Control line from the modem to the terminal stating the data may be sent because the carrier is present. RS-232C is standard.

Cursor BASIC. Find Current Location. See CSRLIN and POS.

Cursor The cursor is the symbol which appears on the screen to let you know where an action (typing in a character, deleting, inserting, etc.) will take place. The cursor moves to the right as you type, backs up to the left when you press the DEL/BKSP key erasing the character to its immediate left. Pressing SHIFT and DEL/BKSP together deletes the character the cursor is on while the cursor remains stationary. Pressing the TAB key moves the cursor over eight spaces to the right.

The Model 100 has four cursor movement keys located on the right end of a row of special keys, above the top row of the standard keyboard. The keys are labeled ←, →, ↑, and ↓ and, when pressed,

will move one space in the direction of the arrow, or, if held down, repeat automatically. In BASIC Entry mode, pressing the ↑ and ↓ cursor keys has no effect, cursor movement being limited to the current line prior to its being entered to BASIC. Backspacing will erase the character the cursor is backspacing over. This is true for BASIC Entry mode, in the SCHEDL and ADDRSS programs, and in TEL-COM entry mode. In each case, once you enter the current line, it cannot be changed. In BASIC Edit mode and the TEXT mode the cursor keys can be used to move the cursor in all four directions without erasing any of the characters it passes over.

In TEXT and BASIC edit mode, the cursor key functions can be expanded by pressing a cursor key and either the SHIFT or CTRL key at the same time. Pressing SHIFT, plus a cursor up or down key moves the cursor to the top or bottom, respectively, of the LCD display. If the cursor is already there, the next screenful in the given direction scrolls onto the LCD. Pressing the CTRL key plus cursor up or down, moves the cursor to the top or bottom of the file, displaying the first or last eight lines, respectively, on the LCD. Pressing SHIFT and the cursor right key moves the cursor to the first character of the next word to the right on the same line. Pressing SHIFT and the cursor left key moves the cursor to the first character of the current word, or if it is already there, to the first character of the next word to the left. Pressing CTRL and a cursor right or left key at the same time moves the cursor to the right or left end of the current line. Remember, these key combinations are only valid in TEXT and BASIC Edit modes. In TEXT and BASIC Edit modes, placing the cursor within existing text allows you to insert characters at the location of the cursor, and define blocks of text (in conjunction with the Function keys) for copying, deleting and moving.

Cursor—Move Left One Space To move the cursor left, use ←, or enter the BASIC statement:

PRINT CHR\$(29)

This only moves the cursor, it does not erase the screen.

Cursor—Move Right One Space To move cursor right, use →, or enter the BASIC statement:

PRINT CHR\$(28)

This only moves the cursor, it does not erase the screen.

Cursor—Move Up One Space To move the cursor up, use ↑, or enter the BASIC statement:

LPRINT CHR\$(30)

This only moves the cursor, it does not erase the screen.

CTRL Key

ASCII Code	CONTROL Key Combination	TEXT Key(s) Equivalent	TEXT Function Performed
1	CTRL A	← SHIFT	Moves cursor to start of first word to left
2	CTRL B	← SHIFT	Moves cursor to last LCD line or last line of next screen if there already
3	CTRL C	SHIFT BREAK/PAUSE	Cancels FIND, SAVE, LOAD, PRINT or SELECT function
4	CTRL D	→	Moves cursor right one character
5	CTRL E	↑	Moves cursor up one line on LCD
6	CTRL F	→ SHIFT	Moves cursor to start of first word to right
7	CTRL G	F3	Saves current file
8	CTRL H	DEL BKSP	Deletes character to left of cursor
9	CTRL I	TAB	Moves cursor to the next TAB stop word to right
10	CTRL J	————	Not used
11	CTRL K	————	Not used
12	CTRL L	F7	Mark the beginning of a text block
13	CTRL M	RETURN	End a paragraph and carriage return
14	CTRL N	F1	Find a string [if after CTRL R stop underline printer control]
15	CTRL O	F5	Copy text block to paste buffer [if after CTRL P start underline printer ctrl]
16	CTRL P	————	Preface printer control character
17	CTRL Q	CTRL ←	Moves cursor to far left of current line
18	CTRL R	CTRL →	Moves cursor to far right of current line
19	CTRL S	←	Moves cursor left one char
20	CTRL T	↑ SHIFT	Moves cursor to top LCD line or top of the next screen if there
21	CTRL U	F6	Delete a block of text, copy it to paste buffer
22	CTRL V	F2	Load a cassette file to TEXT
23	CTRL W	CTRL ↑	Moves cursor to first file character
24	CTRL X	↓	Moves cursor down one line on LCD
25	CTRL Y	PRINT SHIFT	Prints the entire file on printer
26	CTRL Z	CTRL ↓	Moves cursor to the last file character

Cursor, Down • Cycle-Stealing

Cursor, Down (↓) In TEXT and BASIC EDIT modes, pressing the down (↓) key moves the cursor down one line. Holding down the cursor down key causes this process to continue scrolling down the current column through the text.

Pressing SHIFT and ↓ advances the cursor to the last line, same column, of the LCD. Pressing CTRL and ↓ together takes you to the last eight lines of the file, displaying those lines on the LCD and placing the cursor after the last character. See Cursor.

Cursor, Home The Model 100 control code (decimal 11—ASCII) signifies home cursor, returning the cursor to the upper left corner of the screen. In BASIC, entering CLS clears the screen and returns the cursor to the home position. Enter the BASIC Statement:

```
PRINT CHR$(11);
```

See Cursor.

Cursor, Left (←) Pressing the ← key moves the cursor left one character: holding down the ← key causes this process to autorepeat scrolling left along the current line. In all modes except TEXT and BASIC Edit, backspacing the cursor with the ← key erases the characters the cursor passes over.

Pressing Shift and ← moves the cursor to the first character of the current word. If it is already there, the cursor moves to the first character of the next word to the left. Pressing CTRL and ← moves the cursor to the first character of the current line. See Cursor.

Cursor, Right (→) Pressing the cursor right key (→) causes the cursor to scroll right along the current line. When the cursor reaches the rightmost position of the current line it wraps around to the leftmost position of the next line to repeat the process.

In TEXT, and in TEXT as invoked by the BASIC EDIT command, the → key performs additional functions if used in conjunction with the SHIFT or CTRL key. Pressing SHIFT and → moves the cursor to the first character of the next word to the right. Pressing CTRL and → moves the cursor to the rightmost character on the current line. See Cursor.

Cursor, Up (↑) Pressing cursor up (↑) moves the cursor up one line in the same column position on the screen it previously occupied. Holding down the cursor up key will cause this process to continue scrolling up the current column through the text.

Pressing SHIFT and ↑ moves the cursor to the first line, same column of the LCD. See Cursor.

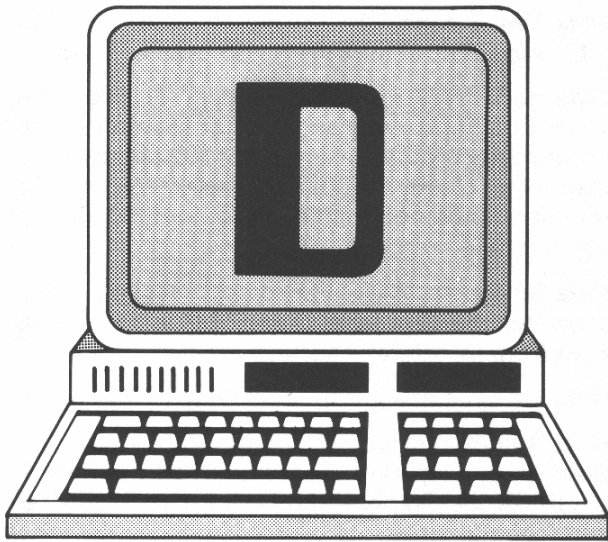
Custom IC Integrated Circuit built to the customer's specifications.

CUTS Cassette User Tape System.

CW Control Word.

Cycle Time Total time a device needs to complete one internal cycle before becoming available again.

Cycle-Stealing Another processor gains access to a microprocessor bus for one cycle. The second processor may be an internal subsidiary processor, as in DMA. Not to be confused with motorcycle-gang activities.



D Codes. ASCII 68, HEX 44. d—ASCII = 100, HEX 64.

D Data line or the hexadecimal symbol for the decimal number 13: D base 16 = 13 base 10 = 15 base 8 = 1101 base 2.

D Flip-Flop A Flip-flop circuit with a delayed reaction. The output is determined by previous input.

D/A Digital to Analog. Conversion from the digital representation used in computers to the analog signals used to drive speakers, motors, etc.

DAA 80C85 Assembly language instruction. Decimal Adjust Accumulator. The eight-bit number in the accumulator is adjusted to form two four-bit Binary-Coded-Decimal digits by the following process: 1) If the value of the least significant 4 bits of the accumulator is greater than 9 or if the AC flag is set, 6 is added to the accumulator: 2) if the value of the most significant 4 bits of the accumulator is now greater than 9, or if the CY flag is set, 6 is added to the most significant 4 bits of the accumulator. All flags are affected. Z, S, P, CY, and AC flags are set.

DAA Data Access Arrangement.

DAC Digital to Analog Converter. A device which converts digital signals (such as from a computer's I/O bus) to analog signals which could control an external device, through varying voltage or current levels.

DAD 80C85 Assembly language instruction. Add Register Pair to H and L. The content of the register pair <rp> is added to the content of the register pair H and L. The result is placed in the register pair H and L. Note: only the CY flag is affected. It is set if

there is a carry out of the double-precision add; otherwise it is reset. The addressing mode is register. The CY flag is set.

Daisy Chain A method of establishing priorities for interrupts. Units capable of interrupting the system can either transmit or block a processor, generated by a unit further away from the processor. The highest priority is given to the unit electrically closest to the processor.

Daisy-Wheel Printer An impact printer which has a wheel with radial spokes bearing type to produce letter-quality output. See Printers for information on particular devices and manufacturers.

Darker Printing Double-Strike Printing. Causes the dot matrix printer to reprint every character with a slight shift down on the page. The dots printed the second time fill in the vertical spaces between original dots creating a darker, more solid character.

To turn on double-strike printing on the Epson RX-80 printer, enter BASIC statement:

```
LPRINT CHR$(27); "G";
```

To return to normal single-strike printing, this mode must be turned off. To turn it off, enter BASIC statement:

```
LPRINT CHR$(27); "H";
```

See Type Formats.

DAS Data Acquisition System.

DATA BASIC Statement. A non-executable statement used in conjunction with the READ statement. It stores the numeric and string constants that are accessed by the program's READ statement(s). The format is:

```
DATA <constant>, <constant>...
```

<constant> may be a numeric or string constant. String constants in DATA statements must be enclosed by quotation marks.

DATA statements are non-executable. They can be placed anywhere in a program. A DATA statement may contain as many constants as will fit on a line, and any number of DATA statements may be used in a program. The READ statements access DATA statements sequentially and read the constant values left to right. The information within a group of DATA statements can be thought of as one continuous list of items, regardless of the number of items on one line.

If the variable type is defined in the READ statement as numeric, the corresponding constant in the DATA statement must be numeric as well. If the variable type is defined as string in the READ state-

Data Acquisition • DATE\$

ment, the corresponding constant in the DATA statement must be a string. See READ.

DATA statements can be "reread" through the use of the RESTORE statement. See READ and RESTORE.

Data Acquisition The collection of data from external sensors, usually in analog form.

Database A systematic organization of data files for central access, retrieval, and update.

Data Bus The set of lines carrying data. All system components are normally connected to the data bus.

Data Concatenation Adding one item to the end of others to produce one longer data or program item.

Data Diskette Used to Store Data. A diskette is called a data diskette when it is used primarily or totally to store data rather than programs. A data diskette usually does not contain many of the DOS programs. Program diskette refers to a diskette which is primarily used to store programs used to process data. A system diskette, on the other hand, is used principally to store DOS commands and related utility-type programs. Often a diskette contains both data and the programs needed to process it.

Data File A file stored on cassette or disk which contains data to be processed by a program. This is a matter of function rather than content. For example, suppose you write a BASIC program SAMPLE.BA. This is a source program. If you then submit it to the BASIC compiler to produce a fast object program, the BASIC compiler treats SAMPLE.BA as its input data rather than as a program to be executed.

A BASIC source program looks more like a document text file than an executable program (.CO). It is a file's use (or usual use) that determines whether it is a data file, a program, or, in the case of a source program, both.

Data File—Kinds and Extensions Information stored on computer-readable media such as data, BASIC programs, part of DOS (such as IBMDOS.COM), or batch files of prepared DOS commands. File is also often used to refer to a collection of data, such as a payroll file with records of information on each of a number of employees. When used in this way, file is short for "data file." The extension (the 1 to 2 letters after the period) of a file name tends to tell you what kind of file it is. See File—Kinds and Extensions.

Data Link Escape An escape character used to introduce control information in a data system.

Data Security Ensuring that data or programs cannot be improperly altered. For example, in an accounts receivable system, steps must be taken to insure that clients cannot alter their invoices. Data security consists of guaranteeing both data integrity and data secrecy or privacy.

Data Separator A circuit in disk controllers which separates the data from the carrier in the signals read from the diskette surface.

Data Set File or a group of related data elements.

Data Tablet Peripheral for graphic input. The position of a stylus on a special sensory surface is digitized. See Graphics Tablets for specific devices and manufacturers.

Data Types A specific interpretation applied to binary data, such as integer, double- or single-precision, real, or string. The default data type for the Model 100 is double-precision numeric. See Numeric Variables, DEFSTR, DEFINT, DEFDBL, and DEFSNG.

Database Management Software The applications programs ADDSS and SCHEDL perform limited database management functions.

Data-Transfer Rate The rate of transfer of data from one place to another, such as from disk to memory or from memory to memory. See also BAUD Rate.

DATA+* Merges files into user-defined formats. The user is prompted for sixteen categories which may be sorted later for precise organization. Files are created for inventory, addresses, appointments, or businesses. This edits, sorts, merges, updates, adds, and deletes files quickly. For home or business. 16K; cassette. Portable Computer Support Group.

DATE\$ BASIC Statement. Used to set or display the system date. DATE\$ holds an 8-character string of the form mm/dd/yy. mm has a valid range of 01 to 12; dd has the valid range of 01-39; and yy represents a two digit number with a valid range of 00-99.

To set the system date use DATE\$ in an assignment statement. The format is:

DATE\$ = "mm/dd/yy"

Two digits must be entered in each of the data segments representing month, day, and year. This means using a leading zero for one-digit dates. Once you enter the system date it will automati-

cally be updated by the operating system when the system clock starts a new day, even when the computer is turned off. This is true as long as the rechargeable ni-cad (nickel-cadmium) battery continues to supply power to RAM memory.

To display the system date use DATE\$ as a variable. For example:

```
OK
PRINT DATE$
09/04/83
OK
```

DAV Data Available. One of the five status bits of a standard UART. Signifies that a character has been received. See PE, OR, TBMT.

DAY\$ BASIC Statement. Used to set or display the system value for day of the week. It may be used as a statement to define the system day. There are seven valid three-letter strings that may be assigned to DAY\$. They are:

MON TUE WED THU FRI SAT SUN

BASIC accepts upper and lower case versions of the DAY\$ variables. To initiate the value of DAY\$, enter the current day of the week in an assignment statement. For example:

DAY\$ = "SUN"

The operating system automatically updates the value of DAY\$ when the system clock begins a new day. That is as the value of TIME\$ goes from 23:59:59 to 00:00:00 the value of DAY\$ goes from MON to TUE, or TUE to WED, etc. This is true even if the computer is off as long as RAM memory continues to be powered by the built-in rechargeable ni-cad battery.

Used as a variable, DAY\$ displays the day of the week currently stored in system memory. For example:

```
OK
PRINT DAY$
SUN
OK
```

DBMS Data Base Management System. Provides a general mechanism for systematic storage and retrieval of data from a data base.

D-Bus Internal destination bus in a CPU, from the ALU to the registers.

DC Direct Current. The Model 100 operates at 6V DC.

DC Motor A motor that operates with a direct current power source, often used in variable speed applications.

DCE Data Communications Equipment. Equipment used with a data communications network—generally, a modem.

DCM Data Communication Multiplexer. Allows several communications processes to operate at the same time over a single channel (such as one phone line). This is possible through time-division multiplexing, which oscillates the channel rapidly from one conversation to another, and through frequency-division multiplexing, which shifts around two different frequencies to simultaneously transmit two conversations. The Model 100's built-in modem uses frequency division multiplexing. The two frequency ranges are 1270/1070 Hz. and 2225/2025 Hz. The frequency range used to transmit the data you send depends upon whether you are answering or calling another computer. When answering, set the range to ANS. The modem transmits data at 2225/2025 Hz., and receives data at 1270/1070 Hz. This is used when your computer is answering another computer. ORIG transmits your data when you dial another computer.

DCO Digitally Controlled Oscillator. An oscillator using a digital circuit to control frequency, rather than an analog circuit (as with a normal oscillator).

DCR 80C85 Assembly Language Instruction. DeCRement Register. The content of register <r> is decremented by one. All condition flags, except CY, are affected. The addressing mode is register. Z, S, P, and AC flags are set.

DCR M 80C85 Assembly Language Instruction. DeCRement Memory. The content of the memory location whose address is contained in the H and L registers, is decremented by one. All condition flags, except CY, are affected. The addressing mode is register indirect. Z, S, P, and AC flags are set.

DCX 80C85 Assembly Language Instruction. Add register pair to H and L. The content of the register pair <rp> is added to the content of the register pair H and L. The result is placed in the register pair H and L. Only the CY flag is affected. It is set if there is a carry-out of the double-precision add; otherwise, it is reset. The addressing mode is register. The CY flag is set.

Deadlock A situation in which two processes wait indefinitely for each other.

Debouncing Eliminating the rapid signal fluctuations which accompany a change of state in mechanical switches. Mechanical springs bounce repeatedly until the contact is finally closed or opened.

Debug or Test a Program • DEFSNG

Typical debounce time for stable contact is 5 to 10 milliseconds. Debouncing may be preformed by hardware (latch) or software (delay).

Debug or Test a Program A programmer must make sure that a program correctly processes all of the types of data for which it is intended. Samples of the data are prepared (test data) and the program is executed using this data (a test run). The program's outputs (reports, screen displays, files, etc.) are then verified to be as specified. An error in the processing logic of a program is called a "bug," hence the terms "debug" and "bug-free."

Debugger An essential program designed to facilitate software debugging. At a minimum, it provides breakpoints, dump facilities, register and memory examine/modify, preferably in symbolic form. See Debug or Test a Program.

Debugging Eliminating the bugs in a program. Troubleshooting and correcting mistakes or errors.

DEC Digital Equipment Corporation, manufacturers of the PDP family of computers.

Decade Counter A counter advancing in increments of ten.

Decode Cycle The second cycle of the fetch-decode-execute sequence of instruction execution. The instruction, contained in the IR, is decoded into a set, or a sequence, of control signals to all the required elements of the system, such as register gates, ALU functions, or external devices.

Decoder A logical unit which decodes two, three, four or more inputs into mutually exclusive outputs. A 3-bit decoder will have eight outputs because a 3-bit number can have eight possible values.

Dedicated Register A register used exclusively to contain a specific item.

Default Parameters The parameter values supplied by a computer system when no explicit values are provided by a program or a programmer.

Default Value A value for a parameter or variable which will be supplied by a program or system, if you do not specify a value. The default value for BASIC variable type in the Model 100 is double-precision numeric.

DEFDBL BASIC. Double-Precision Numeric. Signifies that all variables beginning with the letters in the DEFDBL argument string are double-precision numbers. For example, if you place the statement:

DEFDBL ABC

at the beginning of your program, all variables starting with A, B, or C (such as AXE, BOX, or CUT) will be double-precision numbers because they start with one of the characters defined by the DEFDBL statement to prefix double-precision numeric variable names.

Otherwise, double-precision numbers (numbers with decimal fractions and up to 14 significant digits) can be declared by having their variable names end in #. Note that double-precision numeric is also the default variable type.

Variable names must start with a letter and can have any number of characters but only the first two are significant. They must not be any reserved words, such as IF, ON, THEN, GOTO, etc., or a reserved word followed by a type declaration character (\$, %, !, #). See Reserved Words for complete list. Also see BASIC Variable Names.

DEFINT Integer Variable. If you place the statement:
DEFINT A,B,C

at the beginning of your program, all variables starting with A, B, or C (such as AXE, BOX, or CAT) will be integer whole number variables because they start with a character PREFIX defined as integer by the DEFINT statement. Otherwise, integer variables must end in % and be whole numbers from -32768 to +32767. Integer variable names must start with a letter and can have any number of characters, although only the first two are significant. They must not be reserved words, such as IF, ON, THEN, GOTO, etc., or a reserved word followed by a type declaration character (\$, %, !, #). See Reserved Words and BASIC—Variable Names.

DEFINT BASIC Command. Designates any variables beginning with the letter(s), or range of letters, in its argument list as integer. The formats are:

DEFINT A,B

and

DEFINT L-P

Both argument formats may be used in the same line. If variables with declaration tags, such as O! and LED\$, are used in conjunction with a DEFSTR, they continue to require single-precision and string data respectively, resisting definition as integer variables.

DEFSNG Single-Precision Numeric. Used to define variables as single-precision numbers. For example, if you place the statement:

DEFSNG A,B,C

at the beginning of your program, all variables starting with A, B, or C (such as AXE, BOX, or CAT) will be single-precision numbers because they start

with a character defined by the DEFSNG single-precision variable prefix.

Otherwise, single-precision, floating-point numbers (numbers with decimal fractions and up to six significant digits) can be declared by having their variable names end in !.

Variable names must start with a letter and can have any number of characters although only the first two are significant. They must not be any reserved words, such as IF, ON, THEN, GOTO, etc., or a reserved word followed by a type declaration character (\$, %, !, #). See BASIC Reserved Words, Uses and Restrictions of, and BASIC—Variable Names

DEFSNG BASIC. Designates any variables with the letter(s), or range of letters, in its argument list, as single-precision. The formats are:

DEFSNG R,V,L

and

DEFSNG A-G

Both argument formats may be used in the same line. If variables with declaration tags, such as R# and FUN\$, are used in conjunction with a DEFSNG, they will continue to require double-precision and string data respectively, resisting definition as single-precision variables.

DEFSTR String Variable. If you place the statement:

DEFSTR A,B,C

at the beginning of your program, all variables starting with A,B, or C (such as AXE, BOX, or CAT) will be string variables because they start with a series of characters defined by the DEFSTR statement to be a string variable prefix. Otherwise string variables must end in \$ and contain 0 to 255 letters, numbers, punctuation marks and other characters. String variable names must start with a letter and can have any number of characters, although only the first two are significant. They must not be reserved words, such as IF, ON, THEN, GOTO, etc., or a reserved word followed by a type declaration character (\$, %, !, #). See BASIC Reserved Words, and BASIC Variable Names.

DEFSTR BASIC Command. Designates any variables beginning with the letter, or range of letters, in its argument list as string. The formats are:

DEFSTR X,Y

and

DEFSTR Q-Y

Both argument formats may be used in the same line. Once the DEFSTR is read, string variables may be read into any variable beginning with the letters

on its list. No \$ type declaration tag is needed.

If variables with the declaration tags such as Q! or TOP% are used in conjunction with a DEFSTR, they will continue to require single-precision and integer data respectively, resisting definition as string variables.

Del (“dell”). DElete character in ASCII, a 7F base 16 (hex) or 127 (decimal).

Delay Loop To freeze the screen briefly while the operator using your program reads a message, write a delay loop after you print the message:

1000 FOR Y = 1 TO 2000

1010 NEXT Y

To freeze the screen, put in a dummy input statement and press ENTER. The input variable need not be used in your program:

1000 INPUT “Press ENTER to continue”;A\$

Delete a RAM File To erase any RAM file, BASIC, ASCII, or Machine language you will need to enter BASIC mode. While in BASIC, enter KILL “filename” where filename is the name of the RAM file listed on the Main Menu or on the LCD by the BASIC command FILES. You must include the file extension, .BA, .DO, or .CO, and you must enclose the entire filename in double quotes. Entering KILL “filename” erases the file “filename” from RAM memory.

Delete BASIC Program Lines To delete program line 100, type:

100

and RETURN.

To delete a large number of program lines, it is best to enter EDIT and the range of numbers you wish to delete. This transfers them to TEXT mode where you use the text editing functions to do block deletions.

To delete a block of text in TEXT mode (synonymous with BASIC EDIT mode), place the cursor over the first character to delete and press F7, the Select key in TEXT. Then, move the cursor to the end of the range of text or program lines to delete. The block you have marked will appear in inverse video. Press F6, the Cut key in TEXT to delete the lines. Now press F8, the Exit key, to return to BASIC. The lines you cut while in text mode will be gone.

Delete Character BASIC. If you have not yet pressed ENTER to give the line to BASIC memory, backspace, erasing the characters you backspace over, until you have deleted the unwanted character. Then rekey the portion of the line you erased while backspacing, but do want to keep.

Delete Character • DESC

If you have already entered the line to BASIC memory you have two options. You may rekey the entire line, using the same line number and omitting the character you want to delete. Then enter the line to BASIC memory where it will replace the old line of the same number.

Otherwise, enter EDIT and the number of the line containing the character you want to delete and press ENTER. This translates the line to TEXT where you may use the cursor keys to place the cursor to the immediate right of the character to delete and press the DEL/BKSP key. This deletes the character from the program line. Then press F8, the exit key, to return the line to BASIC memory in its edited form where it replaces the old version of the same line.

Delete Character TEXT. Use the cursor control keys to position the cursor over the character to delete and press the DEL/BKSP and SHIFT keys at once. Alternatively, place the cursor to the immediate right of the character to delete and press DEL/BKSP. In either case, the character is deleted and the remaining text moves in to close the gap.

Delete Current Contents of BASIC Memory Before starting a new program in BASIC, enter:

NEW

This completely erases all program lines currently in BASIC's memory, so if there is something currently in BASIC memory you want to keep, and you haven't already got a copy on RAM memory or on cassette, SAVE it first. If you don't erase the program in memory before starting on another, you will usually wind up with an unuseable combination of mixed lines from your old and new programs.

Delete Current Line BASIC. To delete the current line before it has been entered to BASIC memory press SHIFT and BREAK/PAUSE together.

Delete Last Character Press ← or DEL/BKSP to delete the last character entered. This is valid in BASIC and most applications, except TEXT and BASIC EDIT mode, as long as the line has not yet been entered to the program. In TEXT and BASIC EDIT modes press the DEL/BKSP key at any time to delete the character to the immediate left of the cursor.

Delete Screen BASIC. Clears the LCD screen and returns the cursor to the upper left corner of the screen.

Delete to End of File In TEXT and BASIC EDIT modes press F7, the Select key, to mark the beginning of a text block. Then press CTRL and cursor

down (↓) together to bring the cursor to the end of the file, marking the text block in inverse video. Press F6, the Cut key, to delete the marked text block.

Delete to End of Screen In TEXT and BASIC EDIT modes press F7, the select key, to mark the beginning of a text block. Then press SHIFT and ↓. This moves the cursor to the end of the text block, which appears in inverse video. Press F6, the CUT key, to delete the marked text block.

Delimiter A character which indicates the end of a sequence of characters. In English, the common delimiter of words is the space. See Colon and Period.

DEL/BKSP Located in the upper right segment of the Model 100 keyboard, this key works the same way in all modes. Pressing DEL/BKSP alone backspaces the cursor on the LCD and deletes the character the cursor backspaces over. Pressing SHIFT and DEL/BKSP at the same time deletes the character that the cursor is currently on. In both cases, the remaining text in the file fills in to eliminate the space that was occupied by the erased characters. Holding down DEL/BKSP, with or without the SHIFT, automatically repeats the character-erasing function, so be careful; if you erase data quickly by automatically repeating a key, the LCD may not be able to display the input as quickly as you enter it. None of your key presses are lost, but you won't see the results right away, making it easy to erase too many characters. You may hold down DEL/BKSP for too long and watch helplessly as the cursor continues to delete your text. After doing this a few times, you will begin to develop a sense of timing to avoid this situation.

Demand Paging A dynamic memory management technique which loads disk-resident pages into memory in response to "page faults," i.e., references by programs already in memory to data or instructions that are contained in those pages.

Demultiplexer A logical circuit which can route digital signals from one source to many different destinations. The unit distributes information to many locations or devices in the system.

Density The techniques used to store data twice as densely on a magnetic storage medium, such as MFM, M2FM.

DESC Defense Electronics Supply Center. Manages procurement policies and monitors the quality of military electronics contracts.

Descenders The parts of printed or displayed characters that extend below the baseline.

Development System A system with the capabilities for efficient hardware and software application development for a given microprocessor. Such a system typically includes a microcomputer, monitor, printer, mass-storage (often diskettes or hard disk), PROM programmer, and an in-circuit emulator. Software is often developed on a system totally different from the system on which it will run, either because the target system does not have enough memory or other resources to support development, or because the system is not actually available yet.

Development Tools Hardware and software aids intended for use in developing programs and/or hardware systems.

Device and File Specifiers File specifiers used in commands indicate the type of file being accessed. RAM: and CAS: indicate files saved to RAM memory or cassette tape; they may be accessed at a later date. COM: and MDM: indicate communications files; they are transmitted between the Model 100 and another computer or serial device via the Model 100 RS-232C and built-in modem interfaces. LCL: and LPT: indicate display files; they are routed to the LCD screen or an attached parallel printer for viewing. Output only. With RAM: you must use filename, but the extension is optional. With CAS: the filename is optional (it loads the first file on cassette of the appropriate type). The file extension is optional. No filename is needed with COM:. You must specify a configuration of communications parameters. No filename is needed with MDM:. You must specify a configuration of communications parameters. You must *not* specify baud rate value or <r> parameter. No filename or configuration is needed with LCD:. No filename or configuration is needed with LPT:.

Filename is a six letter (or less) name beginning with a character used with RAM files and cassette files. RAM files have two letter extensions, delimited from the filename by a period. File extensions in RAM are:

- .DO stored in ASCII format
- .BA stored in tokenized BASIC format
- .CO stored as Machine language code.

Device and Ports A port is an address providing a connection between the computer's internal processor and an external device. Ports are used to attach input and output devices. See Port.

Device Names Names which may be used in place of or in conjunction with a file name to specify that data is to come from or go to a device such as the keyboard, CRT, printer, etc. BASIC device names for the Model 100 are:

- CAS: Cassette-stored file
- COM: RS-232C transmission file
- LCD: LCD-Screen transmission file
- LPT: Printer transmission file
- MDM: Modern transmission file
- RAM: RAM memory-stored file

See LOAD, LOADM, MERGE, OPEN, RUN, RUNM, SAVE, SAVEM, and FILE I/O.

DI 80C85 Assembly Language Instruction. Disable Interrupts. The interrupt system is disabled immediately following the execution to the DI instruction. No flags are set.

Diablo A Xerox-owned company which makes computer peripherals and computer systems, and supplies daisy-wheel printers.

Diagnostics A set of routines used to diagnose system malfunctions and/or run standard performance tests.

Die/Dice Circuit elements built of small rectangular pieces of silicon on a wafer. Each wafer may have hundreds of rectangles or dice. Once mounted in a package, it is called a chip.

Digital Having discrete states. Digital logic may have from two to sixteen states. Most logic is binary logic with two states, on or off.

Digital Analyzers Troubleshooting tools which allow the user to locate timing or logic errors.

Digitizer A device which converts analog information to its digital equivalent. Often used for devices which obtain input from a plotting surface and provide coordinates as output.

DIM BASIC. Defines the name and dimensions of one or more arrays. The format is:

DIM <var> <dim>, <var>(<dim>,<dim>,...)...
<var> is the variable name of the array. It may be any type-string, integer, double-precision, etc.
<dim> is a numeric expression, representing a positive number that indicates the number of units in each dimension of the array.

There may be as many dimensions in an array as there is free space available in memory. Because BASIC creates arrays starting with the 0th element, <dim> is less than the number of actual elements in the array.

Diode • Diskette Maintenance

Diode A device which allows current to flow in one direction only.

DIP ("dip"). Dual In-line Package. A standard IC package with two rows of pins, 0.1" apart.

Direct Addressing A technique allowing short instructions, with the address field limited to 8 rather than 16 bits.

Directory of Cassette Files See CLOAD.

Directory of RAM Files The table of contents of a file system which allows easy access to specific files. The Main Menu contains a directory of all RAM files and built-in applications programs. To display a directory of RAM files while in BASIC, enter:

FILES

To get a directory of cassette files, use the CLOAD command with a file name you know is not on the cassette. See SPACE, CLOAD.

Disk A flat, circular, magnetic storage medium which is rotated while in use.

Disk Controller Card A printed circuit board to interface disk storage drives with the CPU of a computer.

Disk File A file on a disk. Also used to refer to the complete disk drive.

Disk Operating System Or, DOS. A collection of programs that helps you manage files and runs the other devices, such as the keyboard and monitor.

Diskette Also called a floppy disk, this flat, circular sheet of mylar substrate coated with a magnetic oxide, rotating inside a protective jacket which continuously cleans the surface.

Diskette, Backup Because diskettes can be made unreadable by physical damage, magnetic contamination, or dirt, it is wise to make a copy, or backup, of any disk containing important information.

Diskette, Bad Sector On A sector on the diskette which will not read/write data correctly, usually due to a minor physical flaw in the diskette. One or two bad sectors will not seriously affect the diskette's use—DOS marks them as bad and avoids using them. More than a few bad sectors indicates that the diskette would make a good frisbee.

Diskette, Data Used to store data rather than programs. In order to save space, a data diskette usually does not have all of the DOS programs. Often, a diskette will contain both data and the programs needed to process the data. See Diskette, Program.

Diskette, Program A program diskette is primarily used to store data processing programs. Often, a diskette will contain both data and the programs needed to process the data. See Diskette, Data.

Diskette, Source The diskette from which information/data is coming. The target diskette is the diskette to which information/data is going.

Diskette, Target The diskette to which information/data is going. The source diskette is the diskette from which information is coming.

Diskette, Write-Protected A diskette is write-protected if it does not have an accessible write-protect notch about one inch down on the right-hand side. The notch could be missing altogether (as in many original software distribution diskettes) or covered over with an adhesive tab. This tab blocks a small spring-loaded switch or a light beam inside the diskette and is sensed by DOS diskette drive programs. You will get an error message (such as "Attempted write-protect violation") anytime you attempt to alter a file on a write-protected diskette by changing it, deleting it, copying a file onto the write-protect diskette, or formatting the diskette.

You are allowed to use, load, or copy files from the write-protected diskette. The purpose of these limitations is to prevent accidental loss of your only copy of programs or data. In most cases, the procedure is to copy the write-protected diskette onto a notched diskette, put away the write-protected diskette as a permanent copy, then modify the notched diskette. In some cases, you will remove the adhesive tab from the write-protect notch and change the original diskette.

It is a good practice to put an adhesive tab (supplied with boxes of diskettes) over the write-protect notch of any important diskette you will backup. Then, if you accidentally ask for the backup in the wrong direction (from the old diskette to your important diskette), you will get a second chance to make the backup rather than losing your data.

Diskette Drive Or disk drive. The machine inside of, or attached to, a computer which turns the diskette, transferring data from the diskette to the computer and vice versa. See Diskette.

Diskette Maintenance File maintenance or diskette maintenance involve keeping track of files on diskettes. This includes creating them, finding them by name, insuring that adequate free space is available on the diskette, maintaining backups, and deleting files no longer needed. These functions

Diskettes by C.R.C. Wholesale • Display Current Function Key Definitions

are supported by various DOS functions, but require thoughtful planning by the user to insure proper results.

Some Database Management Systems attempt to automate part of this work by maintaining files of control and tracking data or providing alternatives to DOS functions.

Diskettes by C.R.C. Wholesale Mini diskettes offered by a distributor of seven major manufacturers of diskettes.

Display A computer output device, such as a CRT or LCD, which displays graphic and/or alphanumeric information.

Display BASIC File Data

Key

I = input to the computer only

O = output from the computer only

I/O = input and/or output

= references the device by using an associated file, the device, or file number; the file name or configuration are associated by the OPEN statement and terminated by the CLOSE statement

@ = default device, redundant to specify this device with this statement or command

[] = not I/O but file related (verify, delete, rename)

Display Current Function Key Definitions Using the LABEL command key displays the function key definitions in the current operation mode. The definitions are displayed on the eighth line of the LCD directly above a row of numbers permanently printed on its lower edge. Each of these numbers corresponds with the function key of the same number. In some built-in application programs, such as TEXT and BASIC, the labels are off when you enter the program from the Main Menu; pressing LABEL turns them on. In the ADDRESS, SCHEDL,

BASIC File I/O Commands and Functions

BASIC Command or Function	RAM:	CAS:	COM:	MDM:	LCD:	LPT:
CLOAD		@ I				
CLOAD?		@ []				
CLOADM		@ I				
CLOSE	# I/O	# I/O	# I/O	# I/O	# O	# O
CSAVE		@ O				
CSAVEM		@ O				
INPUT #	# I	# I	# I	# I		
KILL	@ []					
LINEINPUT#	# I					
LOAD						
LOADM	@ I					
MERGE	@ I					
NAME	@ I					
OPEN	# I/O					
PRINT#	# I/O					
PRINT#USING	# O					
RUN	@ I					
RUNM						
SAVE	@ O					
SAVEM	@ O					

Display Current Memory Contents • DOS

and TELCOM modes the function key label display will be on when you enter the program; pressing LABEL turns it off. In any case, LABEL toggles the function key label display between on and off each time you press it.

Display Current Memory Contents To display specific lines from the program currently in BASIC memory, the format is:

LIST [<line1>[-<line2>]]

<line1> and <line2> are valid line numbers in the range 0 to 65529. <line1> is the first line to be listed and <line2> is the last. A period (.) in place of either line number or alone indicates the current line number, which is the last line number edited, LISTed, or RUN.

You may use a hyphen and one line number to imply a range of lines to list, or two line numbers and a hyphen to specify a bounded line number range.

Three options are available when using the hyphen (-):

If you declare only

LIST <line1>-

that line and all higher numbered lines will be listed.

If you declare only

LIST-<line2>

all lines from the beginning of the program through <line2> are listed.

If you declare both line numbers

LIST <line1>-<line2>

all lines from <line1> through <line2> are listed.

LIST displays all line numbers in the program. Pressing F5, the List key in BASIC, is the same as entering LIST from the keyboard.

To freeze the list while it is being displayed, press BREAK/PAUSE once. To resume listing after a pause press the BREAK/PAUSE key a second time.

Display TEXT File or BASIC Memory Contents on Printer This key has no affect unless the Model 100 is attached to a printer with a standard Centronics-style interface using the Model 100 Printer Cable (RS 26-1409). Pressing PRINT causes the current screen contents, including the function key label display, to be dumped to the printer. Pressing SHIFT and PRINT together prints the contents of the BASIC or text file now open. This also displays the width prompt, followed by the current print width. If you are satisfied with the value given by the width prompt, press ENTER to begin printing. The initial default value is forty characters-per-line,

but once you change that value, it remains as you last specified until you change it again. Valid line width values are between 10 and 132. To interrupt printing before it is finished, press SHIFT and BREAK/PAUSE at the same time.

Display Time See TIME\$.

DLC Data Link Control. Control characters in data transfer used to initiate and terminate communications as well as checking errors.

DLE Data Link Escape. Communications control characters in a data link are distinguished from data characters by the escape character that precedes them.

DMA Direct Memory Access. A method used to provide high-speed data transfers between a peripheral and the main memory. Data is exchanged at maximum memory speed. Several means for accessing the memory are possible. Disconnecting the MPU from the buses is accomplished by a HOLD signal, and requires tristate data and address buses. DMA is performed under the control of a Direct Memory Access Controller (DMAC).

DMAC Direct Memory Access Controller. A device available as a single chip, used to automate DMA transfers. A DMAC is a specialized block-transfer processor which can take bus control away from the MPU and transfers one or more memory words. A typical DMAC can connect to four or eight devices.

DMM Digital Multi-Meter. A volt/ohm meter with digital readout instead of the older needle meter.

DO File Extension. The Model 100 operating system automatically assigns the file extension .DO to all ASCII format files stored in RAM. This applies to document files, created in TEXT mode, as well as BASIC files saved to RAM using the ,A option, which saves the file in ASCII format rather than tokenized BASIC format.

DOD Department Of Defense.

DO-Loop A feature of a high-level language (e.g., FORTRAN) which allows a segment of a program to be executed repeatedly while or until, a certain logical or arithmetic condition is fulfilled.

DOS D-O-S or "doss". A program or collection of programs to control a system whose main secondary storage medium is disk. It usually supplies facilities such as symbolic files, automatic space allocation, dynamic memory allocation, program relocation and loading, utilities, etc.

System allows you to group a number of DOS commands together in a file to accomplish a purpose. These can include the execution commands of programs. In effect, batch processing allows you to define a new DOS command consisting of a sequence of pre-existing DOS commands. It does not allow you to do anything you could not do by typing in the commands one-by-one. Batch processing can, however, eliminate repetitive typing of a sequence of commands. Once a sequence of commands in a BATCH file has been debugged, you can expect it to work every time it is run (provided the data is similar in form each time).

Dot Matrix A method of forming characters by using small dots. Usually patterns are 5 by 7 or 7 by 9, though for very high quality characters patterns of 11 by 13 dots or more are required. This method applies to displays, printers, and other output devices. Characters on the Model 100 LCD are formed by a 5 by 7 dot matrix.

Double-Density Diskette Disk drives can be double-density, single-sided, and double-density, double-sided. They are double-density because the tracks are twice as close together. The only difference is that the double-sided drive has two read/write heads to store information on both the top and bottom sides of the diskette.

Another diskette format is quad- (quadruple) density. Whereas the double-density drives write 48 tracks-per-inch (TPI) of diskette surface, a quad-drive writes twice as close together, or 96 tracks-per-inch. This puts twice as much data within the travel distance of the drive arm.

Double-Precision The default type for BASIC variables is double-precision numeric. To specify a variable as other than double-precision numeric, use a type declaration tag (% , \$, !) or BASIC statement DEFINT, DEFSTR, or DEFNG.

Double-Precision Arithmetic Arithmetic operations which double the precision by using twice as many bits to represent numbers. Double-precision numbers have fourteen significant digits. All arithmetic operations in Model 100 BASIC are done using double-precision arithmetic even when the variable is of another numeric type; it is converted to double-precision for the operation and then converted back to the type of the variable in which it is stored.

Double-Sided Diskette Drives The double-sided drive has two read/write heads to store information on both the top and bottom side of the diskette.

Double-Sided Disk or Diskette A type of disk with both surfaces (sides) used for data storage.

Double-Strike Type Format To set the Epson RX-80 printer in double-strike mode, enter BASIC statement:

`LPRINT CHR$(27);"G";`

To return to normal-strike printing, this mode must be turned off. To do this, enter BASIC statement:

`LPRINT CHR$(27);"H";`

See Type Formats.

Double-Width Type Format To set the Epson RX-80 printer in double-width mode (five characters-per-inch), enter BASIC statement:

`LPRINT CHR$(27);CHR$(14);`

To return to normal printing (ten characters-per-inch), this mode must be turned off. To do this, enter BASIC statement:

`LPRINT CHR$(20);`

See Type Formats.

Dow Jones News/Retrieval* Offers fast, detailed business information and is designed to serve the busy executive or anyone interested in the stock market or other financial news. The service is owned and operated by Dow Jones & Company, Inc., which has a long history of publishing reliable and timely business and financial news.

Both Tymnet and Telenet supporting networks allow connection by most terminals. Hours of operation for Tymnet are between 6 a.m. and 4 a.m. Eastern Time seven days a week. Telenet hours are 6 a.m. to 6 p.m. Eastern Time, Monday through Friday, excluding holidays. Canadian customers can use Datapac.

Items in Dow Jones News/Retrieval are cross-coded in several ways to assist you in finding facts quickly and easily. Examples of the information provided by each data base are as follows: Dow Jones News allows you to access information from over eighty categories on over 6,000 companies in over fifty industries. Also includes over 700 Canadian companies. You can search through headlines and retrieve stories from The Wall Street Journal, Barron's, and the Dow Jones News Service as far back as ninety days.

Dow Jones Quotes/Current delivers quotes on common and preferred stocks, warrants, corporate and foreign bonds, U.S. Treasury Issues, options, and mutual funds.

Dow Jones Quotes/Historical gives monthly summaries back to 1979 and quarterly summaries back to 1978 covering the same subjects as above. Cor-

Down Arrow • Drive, Device

porate Earnings Estimator provides consensus earnings forecasts for 2,400 of the most widely-followed companies compiled from research by over 1,000 analysts at forty-five major brokerage firms.

Disclosure II offers complete company reports such as company profiles and continuously updated information extracted from 10-K, 10-Q, 8-Q, proxy and registration statements for new public companies.

Media General Financial Services gives detailed financial statistics on over 3,200 companies and 180 industries including volume, stock price, and fundamental financial indicators.

Weekly Economic Survey surveys up to fifty of the nation's top financial institutions.

Weekly Economic Update reviews the past week's most significant economic events and gives you a look at the week ahead.

The Wall Street Journal Highlights Online is an electronically delivered summary of *The Wall Street Journal*.

Wall \$treet Week Online gives you the four most recent editions of the popular PBS television program *Wall \$treet Week*.

News/Retrieval Sports Report presents you with all the major scores, stats, standings, and stories for most major sports in college, amateur, and professional fields.

News/Retrieval Weather Report lets you access weather tables for over fifty major cities as well as national summaries and forecasts.

Cineman Movie Reviews reviews fifty current movies, previews coming attractions, and gives ratings from reviewers.

Free Text Search allows you to search for any combination of words, dates, or numbers back to June, 1979 in order to retrieve specific information.

The log-on procedure requires that you connect the computer's built-in modem to a phone line using a modem cable or acoustic coupler, dial the News/Retrieval Access Telephone Number for your area, and respond with the following information to the four computer-generated questions:

Type in your terminal identifier: A and E are the most common

Type in DOW 1;;

Type DJNS and hit the RETURN key

Type in your password and hit the RETURN key

The system will then respond DOW JONES NEWS/RETRIEVAL COPYRIGHT 1980 DOW JONES & CO., INC.

Enter query.

Modems must be compatible with Bell 103, 113, or 212A set at 300 Baud. Dow Jones Information Services.

Down Arrow (↓) See Cursor, Down.

Download a Communications File In TELCOM mode, incoming telecommunications data may be saved, or downloaded, into a RAM file for later use. To initiate downloading, press F2, the Down key. The prompt

File to download?

will appear.

The program is asking for the name of the file that will hold the incoming information. As with any other RAM file name, the first character must be a letter and you may use a maximum of six characters. After keying in the file name, press ENTER and downloading begins. The down label above the F2 function key marker appears in reverse video as long as you use the download function. The incoming data is displayed on the LCD as it goes into memory. To stop downloading press F2 a second time. If you download so much data that the RAM memory becomes full, downloading automatically terminates, retaining whatever portion of the file was received up to that point. After exiting TELCOM mode, you can use TEXT to access and manipulate the contents of the file you have downloaded.

DP Data Processing.

DPDT Double Pole Double Throw switch. A two-polarity switch with two "on" positions.

DPM Digital Panel Meter.

DPMA Data Processing Management Association. Primarily a professional organization for managers of larger computer installations who manage a staff of programmers and/or operators and want to learn about management techniques. For more information, find a copy of *DPMA Journal* at your library or contact DPMA at the address below:

Data Processing Management Association
505 Busse Highway
Park Ridge, IL 60068
312-825-8124

DPSK Digital Phase Shift Keying. Encoding digital data with phase differences on a carrier. See Phase.

DPST Double Pole Single Throw switch. A two-polarity switch with one "on" position.

Drive, Device A mechanical and electrical/electronic device which operates a tape transport or a

floppy disk. It may include several motors for rotation, head positioning, etc., as well as position sensors, control circuits, lights, and switches.

Drive, Diskette A diskette drive (or "disk drive") is the machine installed in or attached to the computer which turns the diskette, reading and transferring data from the diskette to a personal computer and vice versa. See Diskette.

Drive, Source The diskette drive from which information/data is coming. Target drive is the diskette drive to which information/data is going.

Drive, Target The diskette drive to which information/data is going. Source drive is the diskette drive from which information/data is coming.

Driver An amplifier circuit that reshapes the signals on a bus, when more than one transistor-transistor logic (TTL) load is in use.

Drum Rotating magnetic memory, similar to disk, but using the surface of a cylinder.

DS Data Strobe. Enters data into a holding register.

DSR Data-Set Ready (RS-232C standard). A line on a modem, indicating to the data terminal that the received carrier is normal. See RS-232C, DSR, CTS.

DTE Data Terminal Equipment. Equipment which receives or originates data.

DTL Diode Transistor Logic. A logic circuit in which input diodes drive output transistors.

DTR Data Terminal Ready (RS-232C standard). A line on a terminal indicating to the modem that it is ready to send data. See RS-232C, DSR, CTS.

Dual Intensity A printer or display device which can reproduce symbols in boldface or highlight formats, as well as regular intensity.

Dual Processors Two CPUs in one computer. Provides redundancy to increase reliability or productivity, since tasks can be divided and can proceed independently. Also, a system with two different types of processors, with one active or dormant, at any given time.

Dual-Port Memory Memory equipped with dual data and address connections, plus a binary priority circuit. Primarily used for simple communication between multiple processors. Available in single-chip form for small memory sizes.

Dumb Terminal A low-cost data terminal, ordinarily a CRT, which does not have advanced features such as editing keys or local processing.

Dummy Input Statement Use to hold or freeze screen display. To freeze the screen, put in a dummy input statement and press ENTER to proceed. The input variable need not be used in your program:

```
1000 INPUT "Press Enter to continue";A$
```

To freeze the screen briefly, write a delay loop after you print the message:

```
1000 FOR Y = 1 TO 2000
1010 NEXT Y
```

Dummy Variable A stand-in, which will later be replaced by an actual variable name or literal value. For example:

```
<variable> = SQR (<numeric expression>)
```

<variable> and <numeric expression> are dummy variables which must be replaced by actual variables and expressions to produce a valid BASIC statement. See LPOS.

Dump Transfer the contents of one memory device to another. Internal registers may be dumped to memory, memory can be dumped to disk, printer, or screen, etc.

Dump Screen to Printer In any application, pressing the PRINT command key copies the current display from the screen to the printer providing a one-time dump of the entire screen.

Duplex Bi-directional communication method allowing simultaneous data transfers in both directions. Can use separate lines or multiplex a single line. When in TELCOM terminal mode, you can enter data directly to a host computer, or send a previously created TEXT file. The characters are displayed on the screen as they are transmitted, and their meaning depends on whether the system being used is half or full duplex. Full duplex echoes the information that the host computer has received on the screen so that you can make sure that it was not altered during transmission by a "noisy" phone line. Half duplex only displays what you enter, not what is transmitted, so it is impossible for you to tell if they were received by the host computer intact. In TELCOM terminal mode F4, the Full key lets you toggle between full and half duplex.

DUT Device Under Test.

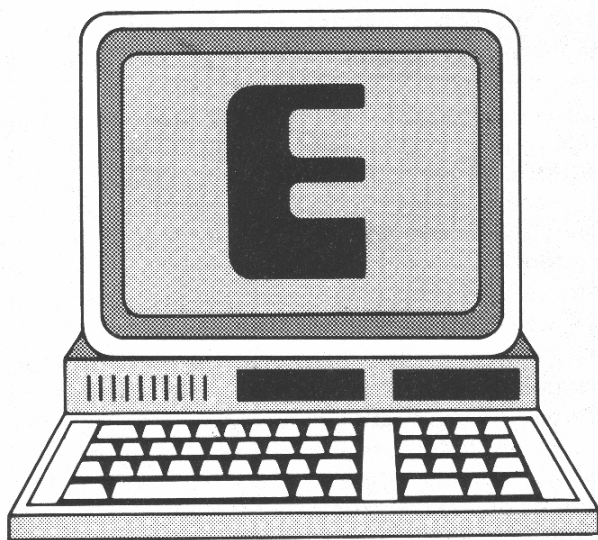
Dyadic An operation employing two input values, such as addition or multiplication. Contrast to a monadic operation with one input value like squaring.

Dynamic Circuitry which stores information as charges on MOS capacitors. Usually volatile, requiring periodic refreshing.

Dynamic Memory • Dynamic Memory Allocation

Dynamic Memory MOS RAM memory using dynamic circuits. Each bit is stored as a charge on a single MOS transistor. This allows very high density (only one transistor is used per bit). Since the stored charge leaks, a typical dynamic memory must be refreshed every 2 milliseconds by rewriting its entire contents. This does not slow down the system, but does require additional memory refresh logic. Dynamic memory chips are cheaper than static ones, and are generally preferred for memory sizes over 16K. Dynamic memory is also volatile, i.e., the data stored is lost when power is turned off or interrupted.

Dynamic Memory Allocation Allocating memory to multiple concurrent programs according to their needs. A strategy for optimizing performance.



E Codes. ASCII 69, HEX 45. e—ASCII 101, HEX 65.

E Enable. Also, the hexadecimal symbol for the decimal number 14: E base 16 = 14 base 10 = 16 base 8 = 1110 base 2.

e Natural Logarithms. See EXP

EA Electronic Arrays.

Earphone Outlet A port on most cassette recorders. See *Cassette Cable, Installing*, *Cassette I/O BASIC Command Table*.

EBCDIC (“ep-si-dick”). Extended Binary Coded Decimal Interchange Code. The 8-bit code used by IBM to encode their character set, based on the original punched card codes. Encodes essentially the same characters as ASCII, but with different bit patterns.

E-Beams Electron Beams.

Echo A character received from the keyboard is sent to the printer or screen for display.

Echo—Hardware Testing A loop-back or “Echo” technique is used to test the circuits of an input/output device by “looping” whatever is sent out back to the computer as if it were input. In this way, the circuits inside the computer are tested in isolation from the circuits of the external device to locate a fault.

Echo-keyboard to screen Echo describes the process of sending characters typed on the keyboard to the screen for a visual confirmation of what has been typed. There are no hardwired connections between the keyboard and the screen; the keyboard simply enters characters into memory. The ROM BIOS programs of PC DOS then copy the

characters from memory to the screen, creating a duplication or “echo” of what was typed.

ECL Emitter Coupled Logic. A bipolar transistor-transistor logic (TTL) device using variations in the current, rather than voltage levels, to control bit and gate switching. While faster than other circuits, ECL-TTL circuits have a much higher power consumption and heat production.

ECM Electronic Counter Measures. Using electronic devices to jam or evade an opponent’s electronic equipment. Used in military and espionage situations.

Edge Card Connector An edge of a printed circuit board which has exposed conductors (“ways”) to form an electrical connection with another board through a slot.

EDIT BASIC Command. The EDIT command is used to translate a single line, range of lines, or entire program from encoded BASIC format to ASCII format so that it can be edited using the TEXT mode. The format is:

EDIT <numrng>

<numrng> is the line number or range of line numbers you want to edit. There are four possible configurations for <numrng>. They are:

1) A designated range, line number to line number inclusive. For example:

EDIT 10-30

translates lines 10 through 30 to TEXT mode.

2) An open ended, implied, range. For example:

EDIT 30-

translates to TEXT all lines numbered 30 and greater, up to the highest and last line number in the program. Similarly the other version of the implied range.

EDIT -300

will translate lines 30 and less, down to the lowest and first number in the program, to TEXT mode.

3) The last line number accessed. For example:

EDIT

translates the last line used (listed, entered, edited, etc.) to TEXT mode.

4) The entire program. Here no given line number range implies the full range. For example:

EDIT

In this case, BASIC translates the entire program to TEXT mode.

Once you enter TEXT mode from BASIC using a version of the EDIT command, you continue to be in TEXT mode until you exit back to BASIC by press-

Edit—Change Contents of a File • Editor

ing F8, the Exit key in BASIC EDIT (i.e. TEXT) mode. This translates the ASCII text back into tokenized BASIC in BASIC memory.

If you try to translate text back to BASIC by pressing F8 (violating BASIC program syntax by not starting with a line number), you will hear a beep sound and see the message:

```
TEXT ILL FORMED
PRESS SPACE BAR FOR TEXT
```

You won't be able to leave TEXT mode and return to BASIC until you supply line numbers for every line.

The portions of the program remaining in BASIC after you translate some specified program range to TEXT mode are vulnerable to everything. That is, you may create a new line number while in TEXT (EDIT mode) that already exists in the segment of the program still in BASIC. If you then translate the text you have been editing back to BASIC, with the F8 key, the new line will replace the old line with the same line number still in BASIC.

Any changes made to a program in BASIC memory using EDIT is reflected in the copy of that program stored in RAM with a .BA extension. If you want to keep a pre-edited version of your program, save the program to RAM as an ASCII file using the A option of the SAVE command. You may then edit the program in BASIC memory and those changes will not be reflected in the copy of the program stored in RAM with the .DO extension.

Edit—Change Contents of a File The Model 100 stores three kinds of files in RAM memory. The Model 100 text editor, TEXT, will only edit ASCII format files. However, you may temporarily translate BASIC files currently in BASIC memory to ASCII format using the BASIC EDIT command.

To edit a RAM document file, .DO, place the cursor on the Main Menu over the name of the file to edit and press ENTER. This displays the first eight lines of the file on the LCD and enables TEXT mode. Alternately, place the cursor on the Main Menu over TEXT and press ENTER. When you see the prompt File to edit? on the LCD, enter the name of the document file in RAM you want to edit and press ENTER. The .DO extension is optional. This also displays the first eight lines of the file on the LCD and enables TEXT. You may now use all the text editing functions of TEXT mode on the file. When you are finished editing, press F8, the Menu key in TEXT. This saves your ASCII file, with all the changes you have made, in RAM, using the same file name you entered to begin editing. It also returns you to the Main Menu.

To edit a RAM BASIC file, .BA, place the cursor on the Main Menu over the name of the file to edit and press ENTER. This automatically executes, or turns the BASIC program. Break program execution by pressing the SHIFT and BREAK/PAUSE keys at the same time. The prompt OK indicating that you are in BASIC command mode then appears. The BASIC file you entered is now in BASIC memory. Check this by typing ENTER LIST. (The program lines form the BASIC file that is listed on the CCD).

Alternately, you may place the cursor on the Main Menu over BASIC and press ENTER. The prompt OK appears, indicating that you are in BASIC command mode. To get the BASIC RAM file to edit into BASIC memory, press F2, the Load key in BASIC. When you see the prompt LOAD" on the LCD key, enter the name of the file to edit. Otherwise, simply type ENTER LOAD" and the file name. Your BASIC file to edit is then in BASIC memory, as you will see if you type and enter LIST. (The program lines will list on the LCD).

To translate the tokenized file in BASIC memory into ASCII format for editing, enter the BASIC command EDIT. This translates the entire contents of BASIC memory, your tokenized BASIC file, into ASCII format, placing it into a separate, temporary RAM file. The first eight lines of your BASIC file then appear on the LCD, and TEXT mode will be activated.

When you are finished editing the BASIC program, press F8, the Exit key. This translates the program lines in your temporary edit-file from ASCII format back to tokenized BASIC in BASIC memory. Now, if you LIST your program, you will see the changes you made in TEXT mode. These changes are also automatically reflected in the BASIC file in RAM which you originally loaded to BASIC memory for editing. There is no need to save the program back to RAM. You can check this by exiting BASIC, F8, which erases BASIC memory contents, and reentering your RAM program file into BASIC memory using one of the methods discussed above. When you list the file you will see that your editing changes are indeed there. See EDIT.

Editor A program which facilitates the entry and maintenance of text in a computer system. Typical operations include inserting, deleting, copying, and searching for words, lines, or blocks of text. BASIC edit and text processors such as TEXT mode in the model 100 provide editing capability for any line displayed on the screen (full-screen editors). Their basic function is to create and change text data such as a letter, report, program, or book.

Editor, Text An editor program specialized for text files. A text editor manipulates ASCII characters such as letters, punctuation marks, etc. TEXT and the BASIC editor are text editors. Their basic function is to create and change text data such as a letter, report, program or book. See TEXT Mode, BASIC EDITing.

EDP Electronic Data Processing. Processing data with electronic machines such as adding machines, calculators, and computers.

EFL Emitter-Follower Logic. A transistor circuit in which the base element is common to the output and input circuits.

EI 80C85 Assembly Language Instruction. Enable Interrupts. The interrupt system is enabled following the execution of the next instruction. No flags are affected.

EIA Electronic Industries Association.

EIA-RS-232C The EIA standard for serial data-transmission interfaces in asynchronous communications. Data is sent in 10- or 11-bit serial bundles. The first bit is called the start bit. It signals the beginning of the data. The data bits follow, from least to most significant. Following the last data bit comes the stop bit. The RS-232 interface converts data bytes from the parallel form used internally by the Model 100 to the serial form used in computer-to-computer communications and vice versa. See TELCOM and File I/O Commands and Functions.

Electron The elementary unit of negative electrical charge. Often conceived as a particle circling around the nucleus of an atom. Electrical current is a flow of electrons.

Electron-Beam A straight (rendered in parallel waves) beam of electrons used in manufacturing ICs and in CRT display systems.

Electronic Disk See RAM Disk.

Elephant Floppies The EMS #2 is a 5¼ inch, single-sided, double-density, soft sector disk with a hub ring, certified 100% error-free. Leading Edge Products, Inc.

ELINT Electronic INTElligence.

EMI Electro-Magnetic Interference. Interference caused by electrical fields produced by capacitive coupling, magnetic fields produced by mutual inductance, or electro-magnetic fields (radio waves).

Emulation Simulation in real time. One computer emulates another computer by executing an emulator program that makes it interpret the same instructions.

Enable To make a device ready or available to function. Opposite of disable.

END Statement marking the end of a program in several programming languages. See END.

END BASIC Statement. END is used to terminate program execution, close all files, and return to command level. The format is:

END

END statements may be placed anywhere in a program you wish to terminate execution. END differs from STOP in that it does not cause a "Break" message to be printed, and it closes all files. An END statement at the end of a program is not imperative. If you have used END to terminate execution before the logical end of the program, the last line number, you may restart execution with the next line number by entering the BASIC command CONT.

END functions differently with the BASIC Compiler. See Compiler Manual.

End—Keyboard To end a BASIC program from the keyboard, press the BREAK/PAUSE and SHIFT keys at the same time. You will see the message Break in <xxxx> on the LCD; <xxxx> is the line that was executing at the time of the keyboard interrupt. To resume execution at the place it was interrupted, enter BASIC command CONT. See Control Keys.

End Current Line Press ENTER to end the current line, send the line to the requesting program, and put the cursor at the start of the next line. This is valid under BASIC and in most application programs. In TEXT mode, pressing the ENTER key signals the end of a paragraph or block of text and may be used to start a new line on the LCD before the forty-character automatic wrap length has been reached.

End of File, Delete to In TEXT and BASIC Edit modes, press F7, the Select key, to mark the beginning of a text block. Then press CTRL and the Down (↓) key together to advance the cursor to the end of the file, marking the text block in inverse video. Then press F6, the Cut key, to delete the marked text block.

End of Screen, How to Get to In TEXT and BASIC EDIT modes, pressing the SHIFT and Down ↓ keys moves the cursor to the end of the current screen.

End of Text, How to Get to In TEXT and BASIC EDIT modes, pressing the CTRL and Down ↓ keys together advances the cursor to the last character in the file.

ENQ • Erase Character

ENQ ENQuiry control character.

ENTER Key Carriage Return. Pressing ENTER ends the current line, gives the line to the requesting program, and positions the cursor at the start of the next line. This is valid in BASIC and most other application programs. In TEXT mode, pressing the ENTER key signals the end of a paragraph or block of text and may be used to start a new line on the LCD before the forty-character automatic screen wrap length has been reached.

Entry Mode—TELCOM When you first enter TELCOM, the program is automatically in Entry mode. You can use it to dial the phone so you can talk to another person or so the computer can communicate with another computer. You can also bypass the phone lines and the modem to send data directly to another computer attached to the Model 100's RS-232C port. Entry mode assists in establishing a communications link in three ways: by retrieving a phone number and perhaps other access information from the ADRS.DO file; by selecting the computer to computer communications parameters to be used and the type of communications device—modem or RS-232C; and, if used with the direct connect modem cable, by automatically establishing the phone connection, autodialing, switching to Terminal mode, and logging on to a host system (if ADRS.DO entry is set up to do that).

Environment The state of all registers, memory locations, and other components and conditions in a system. Also used to refer to a software environment, such as DOS.

EOB End Of Block.

EOC End Of Character, or End Of Conversion for an ADC.

EOF BASIC Function. EOF function indicates when the end-of-file has been reached. The format is:

variable = EOF<filenum>

<filenum> is the file, or buffer, number assigned to the file in the OPEN statement.

The EOF function can be used to tell if the end of a specified file has been reached, and to avoid an "Input past end" error. EOF returns -1 (true) if end of file has been reached on the specified file. A 0 (zero) is returned if end of file has not been reached. A -1 for a communication file means that the buffer is empty.

EOF End Of File.

EOR Exclusive OR, (XOR). Also, Electro-Optical Reconnaissance. Gathering information through electrical and optical surveillance. Used in scientific studies, military, and espionage applications.

EOT End of Transmission

EPROM ("ee-prom"). Erasable Programmable Read-Only Memory. A PROM which can be reprogrammed several times. Typically, an ultraviolet-erasable PROM which can be erased by exposing it for several seconds to hard ultraviolet light. It is then reprogrammed with a special PROM programmer, and retains its contents for years. A UV-erasable EPROM may be recognized by the quartz window over the chip. Other EPROMs are electrically erasable.

Epson Printer BASIC. See Type Formats to learn how to set the Epson RX-80 printer's print size, strike method, or number of lines-per-inch.

Equal Sign (=) See BASIC Relational Operators Table.

Erase All BASIC Program Lines Before starting a new program in BASIC, enter:

NEW

This completely erases all program lines in BASIC's memory, so SAVE anything in BASIC memory you want to keep. If you don't erase the program in memory before starting another, the old and new program lines will mix, making your program unusable.

Erase BASIC Program Lines To delete program line 100, enter:

100

and press RETURN.

To delete a large number of program lines, it is best to enter EDIT, and the range of numbers to delete. This transfers them to TEXT mode where you may use the text editing functions to do block deletions. To delete text in TEXT mode (synonymous with BASIC EDIT mode), place the cursor over the first character to delete and press F7, the Select key. Move the cursor to the end of the range of text to delete, then press F6, the Cut key, to delete the marked lines, which appear in inverse video. Press F8, the Exit key, to return to BASIC.

Erase BASIC Variables See NEW, CLEAR, LOAD.

Erase Character Position the cursor over the character to delete and press the DEL/BKSP and SHIFT keys at the same time. Or, place the cursor to the immediate right of the character to delete and press the DEL/BKSP key.

Erase Characters BASIC. To erase characters before saving them to BASIC memory, backspace over letters until you delete the unnecessary portion of the line. You must retype the portion of the line you erased while backspacing, but do want to keep.

If you have already entered the line to BASIC memory, you have two options. You may retype the entire line, using the same line number and omitting the character you want to delete, and save the line. Or you may transfer the line to TEXT, and delete the character while in the editing mode. Enter EDIT, the number of the line containing the character you want to delete, and press ENTER. Use the cursor keys to delete the unwanted character from the program line, then press F8, the Exit key, to replace the line in BASIC memory.

Erase Current Line BASIC. Pressing the SHIFT and BREAK/PAUSE keys at the same time prints a "c" on the screen at the end of the line you wish to delete, and does not enter it to memory. It will not appear when you list the program.

Erase File from RAM To erase any BASIC, ASCII, or Machine language RAM file, you need to enter BASIC, then type KILL, the name of the RAM file you wish to erase, the file extension (.BA, .DO, or .CO), enclosing the entire file name in double quotes.

Erase Last Character Press ← or DEL/BKSP to delete the last character entered. This is valid in BASIC and most applications, except TEXT and BASIC EDIT modes, as long as the line has not yet been entered to the program. In TEXT and BASIC EDIT modes, press the DEL/BKSP key to delete the character to the left of the cursor.

Erase Screen BASIC. CLS clears the screen and returns the cursor to the upper left corner of the screen.

Erase to End Of File In TEXT and BASIC Edit modes, press F7, the Select key, to mark the beginning of the text to be deleted, press CTRL and the Down ↓ key at the same time to advance the cursor to the end of the file, marking the text block in inverse video. Press F6, the CUT key, to delete the marked text block.

Erase to End of Screen In TEXT and BASIC EDIT modes, press F7, the Select key, to mark the beginning of a text block. Then press SHIFT and ↓ to move the cursor to the end of the text block, which will then appear in inverse video, and press F6, the Cut key, to delete the marked text block.

Erasing All RAM Memory Contents See Cold Start.

ERR and ERL BASIC Variables. ERR contains the error code for the last error, and ERL contains the line number of the line in which the error was detected. The formats are:

ERR

and

ERL

The ERR and ERL variables are usually used in IF...THEN statements to direct program flow in the error handling routine. See ON ERROR...GOTO. If you test ERL in an IF...THEN statement, be sure to put the line number on the right side of the relational operation. For example:

IF ERL = <line number> THEN...

If the error was caused by a mistake in I/O, ERL will be 23. You could create an error, halt execution, and print an error message on any other error. For example:

100 ON ERROR GOTO 200

.

200 IF ERR = 23 THEN RESUME ELSE PRINT
"ERROR"; "IN LINE"; ERL: STOP

ERR and ERL can also be set using the ERROR statement. See ERROR Code Numbers ERROR, ON ERROR...GOTO.

Error BASIC. See ERR and ERL

ERROR BASIC Statement. Simulates the occurrence of a BASIC error, and allows you to define your own error codes. The format is:

ERROR<numex>

<numex> must be an integer expression between 1 and 255.

The Model 100 has error messages which are referenced by certain codes. If the value of <numex> is the same as an error code used by BASIC, the ERROR statement simulates the occurrence of that error. If an error-handling routine has been defined by the ON ERROR statement, program execution continues to the error routine. Otherwise, the error message corresponding to the code is displayed, and execution halts.

To define your own error code, use a value that is different from any used by BASIC. BASIC uses error code numbers one through fifty-eight, displaying its own two-character error message for each one. You may define your own error code for undefined error code numbers 21, 34-49, and 59-225. Your

Error Code Number of Last Error • ESC

new error code may then be tested in an error handling subroutine which can be set to print a special error message, resume execution, etc.

If you create your own code, but not an error-handling routine for it, an "Unprintable error" message is displayed and execution halts. See Error Code Numbers, Error Messages, ON ERROR...GOTO, ERR, and ERL.

Error Code Number of Last Error BASIC. See ERR.

Error Code Numbers BASIC.

Error Number	Error Message	Error Condition
1	?NF Error	NEXT without FOR
2	?SN Error	Syntax Error
3	?RG Error	RETURN without GOSUB
4	?OD Error	Out of Data
5	?FC Error	Illegal Function Call
6	?OV Error	Overflow
7	?OM Error	Out of Memory
8	?UL Error	Undefined Line
9	?BS Error	Bad Subject
10	?DD Error	Double Dimensioned Array
11	?/0 Error	Division by Zero
12	?ID Error	Illegal Direct
13	?TM Error	Type Mismatch
14	?OS Error	Out of String Space
15	?LS Error	String Too Long
16	?ST Error	String Formula Too Complex
17	?CN Error	Can't Continue
18	?IO Error	Error
19	?NR Error	No RESUME
20	?RW Error	RESUME Without Error
21	?UE Error	Undefined Error
22	?MO Error	Missing Operand
23-49	?UE Error	Undefined Error
50	?IE Error	Undefined Error
51	?BN Error	Bad File Number
52	?FF Error	File Not Found
53	?AO Error	File Already Open
54	?EF Error	Input Past End of File
55	?NM Error	Bad File Name
56	?DS Error	Direct Statement in File
57	?FL Error	Undefined Error
58	?CF Error	File Not Open
59-255	?UE Error	Undefined Error

Error Correcting Code A data storage or transmission code using extra bits which automatically detect and correct single- or multiple-bit errors.

Error Correction Methods used to correct erroneous data produced by defective or unreliable data storage and transmission systems.

Error Message A statement or code printed out or displayed on the screen by a program to let you know something is wrong.

Error Messages BASIC.

Error Message	Error Number	Error Condition
?AO Error	53	File Already Open
?BN Error	51	Bad File Number
?BS Error	9	Bad Subscript
?CF Error	58	Closed File Accessed
?CN Error	17	Can't Continue
?DD Error	10	Doubly Dimensioned Array
?DS Error	56	Direct Statement in File
?EF Error	54	Input Past End of File
?FC Error	5	Function Call Illegal
?FF Error	52	File Not Found
?FL Error	57	Undefined
?ID Error	12	Illegal Direct
?IE Error	50	Undefined
?IO Error	18	Error
?LS Error	15	String Too Long
?MO Error	22	Missing Operand
?NF Error	1	NEXT without FOR
?NR Error	19	No Resume
?NM Error	55	File Name Bad
?OD Error	4	Out of Data
?OM Error	7	Out of Memory
?OS Error	14	Out of String Space
?OV Error	6	Overflow
?RG Error	3	RETURN without GOSUB
?RW Error	20	Resume without Error
?SN Error	2	Syntax Error
?ST Error	16	String Too Complex
?TM Error	13	Type Mismatch
?UE Error	21,23-49, 59-255	Undefined Error
?UL Error	8	Undefined Line

ESC ESCape. Causes the terminal and/or processor to interpret subsequent characters differently. Escape codes are used to indicate a sequence of control messages in ASCII. For example, ASCII 27 is an escape code to the EPSON dot matrix printer to interpret one or more following bytes as control information rather than data to be printed. See Type Formats.

ESC Press the escape key twice to exit from TEXT mode. This is synonymous to pressing F8, the Menu key.

ETB End of Transmission Block

Ethernet An inter-computer communications network developed by Xerox Corporation.

ETX End of TeXt.

EUROMICRO European Association for Micro-processing and Microprogramming.

Even Parity Even Parity adds one bit to all odd-numbered bit patterns to make them even.

Excess 3 Code. A variation of BCD which uses binary values of three through twelve to represent the decimal integers 0 through 9.

Execute Start, or run, a program. In BASIC, when you enter RUN, you execute the program in memory.

When the computer is in Execute mode, it is interpreting and acting on the BASIC keywords in a program or command line. You cannot enter characters from the keyboard while BASIC is running unless BASIC requests them with the prompt, "?," and information regarding the data to enter.

To temporarily pause program execution, press the BREAK/PAUSE key. To resume execution at the program line where it was paused press the BREAK/PAUSE key a second time. The computer will not respond to any key pressed while program execution is paused in this way.

To break program execution, press the SHIFT and BREAK/PAUSE keys at the same time. The computer will display the "Break" message in the program line that was executing at the time of the break, and enter the Command mode. Use command lines (such as print and assignment statements) to display and change variable values. To resume program execution with the line that was executing at the time of the break, enter CONT. Program execution will not resume if you have altered any program lines during the break.

See Automatic Program Execution, LOAD, RUN, RUN a BASIC Program, Run a Machine Language Program.

Execute BASIC Program See RUN and LOAD, Run a BASIC Program, Automatic Program Execution.

Execute BASIC Program Automatically Positioning the cursor over the name of the RAM file on the Main Menu containing an encoded BASIC program—indicated by the file extension .BA—and pressing ENTER automatically runs the program. It is also possible to autorun a BASIC program before you turn the computer off so that it will autorun the next time you turn the computer on.

First enter BASIC and write or LOAD the program you want to autorun. SAVE the BASIC program to RAM and enter:

IPL <filenm>

where <filenm> is the name you gave to the BASIC file to autorun. Now, turn off the computer. The next time you turn it on, your BASIC program will automatically execute. See IPL.

Execute Cycle The third of three cycles for program instruction execution. During this cycle the actual operation is performed. See Fetch Cycle and Decode Cycle.

Executable Files The extension (the one or two letters after the period) of a file name tends to tell you what kind of file it is. For example, .CO is the extension for a Command file.

Execution Time The time required to execute an instruction, including fetch-decode-execute. Also used to refer to the point in the processing of a program when it is given control of the CPU, as contrasted with compile time, link time, and load time.

Exerciser A test system or program designed to detect malfunctions in memory, disks, tapes, or other devices prior to shipping.

EXP BASIC Function. Calculates the natural antilog (the exponential function which is the mathematical number "e" raised to the x power). "e" is the base for natural logarithms. If x is outside the range +87.33.65, overflow occurs.

Expansion Board A printed circuit board which accommodates extra components or cards for the purpose of expanding a computer. It is called an expansion chassis if it connects an additional cabinet to the system.

Expense Register* This software can handle records for 300 transactions on 15 accounts. Printouts may be generated on your data by date or account code, and you may save your files to RAM or cassette. 32K. L/R Software.

EXPN+* Maintains detailed records of expenses and financial information, and creates budget, personal or departmental cash flow, and project reports. Eighteen categories and twelve days construct the chart format. Total expenses for each day are accumulated upon request. This is an extremely flexible program for maintaining precise, detailed records of financial activities. Part of BusinessPak+. 16K; cassette. Portable Support Group.

Exponential BASIC. See EXP.

Extended Arithmetic Double-precision operations. Also used to refer to expanded capabilities, such as built-in trigonometric functions.

Extension and File Name A file name is one to six characters in length with an optional device name, separated by a colon. An optional two-character file extension may follow, separated by a period. The entire specification for a file consists of the three parts <device>:<filename>.<extension> and is often called a <filespec> in documentation. Example:

RAM:BOOKS.DO

Normally, only RAM files have extensions which are automatically assigned by the operating system when they are saved to RAM. File extensions are .DO—an ASCII file; .BA—an encoded BASIC file; .CO—a Machine language file. Devices on the Model 100 are:

CAS for cassette tape stored file; COM for RS-232C transmission file; LCD for LCD screen transmission file; LPT for printer transmission file; MDM for modem transmission file, and RAM for RAM memory stored file.

See LOAD, LOADM, MERGE, OPEN, RUN, RUNM, SAVE, SAVEM, and File I/O.

External Device All external devices talk to the computer through address ports. Ports are used to attach input and output devices. See Port.

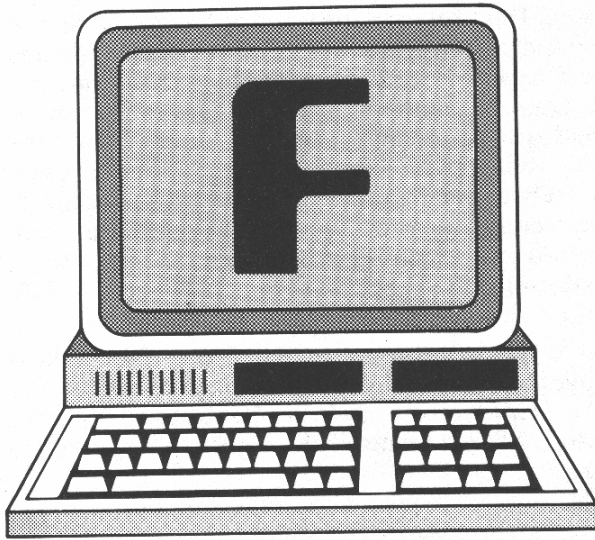
External Memory Memory is any device which can store information and allow it to be retrieved when needed. The Model 100 relies primarily on Random Access Memory (RAM), Read-Only Memory (ROM), cassettes, and hard disks. Memory, though, is usually a reference to RAM, the general purpose, erasable-reusable memory located inside the Model 100.

ROM contains fixed data—programs such as the Model 100 BASIC and the built-in applications programs. ROM BIOS contains the Machine language programs which run the devices attached to the computer. ROM and RAM together make up the internal memory or main memory of the Model 100, or any other computer.

External memory, such as cassettes, diskettes, and hard disks, involves mechanical motion to retrieve data and is thus hundreds or thousands of times slower than internal memory, while data in internal memory is immediately available. Data in external memory must be copied into internal memory (READ or INPUT), processed, then copied back out to external memory (WRITE or OUTPUT).

If data has been created, it can be written out to external memory without a READ first. And, if data read in from external memory has not been modified, there is no need to write it back out since the original copy is still there.

While external memory is much slower than internal memory, it also has advantages. It is much cheaper per character of data stored on-line (available for processing without manual intervention). In addition, the ability to store external memory data off-line (such as diskettes or cassettes in a box) allows essentially unlimited storage of data. This, of course, requires that you must load the diskette, cassette, or disk before the data can be read into the internal memory for processing. See also Virtual Memory, RAM Disk, Spooling, Bank Switching, Memory Maps, Memory Addresses, Memory Segments, Bubble Memory, RAM Memory Cards, Diskette Formats.



F Codes. ASCII = 70, HEX = 46. f—ASCII = 102, HEX = 66.

F Flag. Also, Finish in BNPF code, or the hexadecimal symbol for decimal 15—the largest hexadecimal digit: F base 16 = 15 base 10 = 17 base 8 = 1111 base 2.

F-8 Fairchild's 8-bit microprocessor.

Fairchild Major semiconductor manufacturer located in Silicon Valley (between San Francisco and San Jose, California).

Fan-Fold Paper Continuous sheets of paper joined along perforations and folded in a zigzag fashion. Often used with printers because it can be continuously fed and folded without operator assistance.

Fan-In An electrical load presented to an output by an input.

Fan-Out The electrical load that an output is capable of driving. Ordinarily expressed as the number of inputs that can be driven from a given output signal.

Farad The unit of electrical capacitance. A one volt-per-second change in voltage across a one farad capacitor will require one ampere of current flow.

Fast Fourier Transform An application of the Cooley-Tukey algorithm to Fourier transforms which allows computation of Fourier transforms with dramatically reduced time and storage requirements.

Fatal Error A condition occurring during the execution of a program which requires termination of the program.

Fault-Tolerant A program or system capable of correct operation even when one or more of its components have failed. Also called error-tolerant or fail-soft.

FCB File Control Block.

FCC Federal Communications Commission.

FD Floppy Disk.

FDC Floppy Disk Controller.

FDM Frequency Division Multiplexing. This technique assists in carrying data from a number of terminals simultaneously by dividing a high band width into several narrow bands of lower frequency.

FE Framing Error. One of the five status bits of a standard UART. It becomes true if the incoming character lacks a valid stop bit. See PE, OR, DAV, and TBMT.

Feedback When one or more outputs of a system is also used as input in a control loop.

FET ("fett") Field Effect Transistor. A tiny, low-power transistor such as a MOS (Metal Oxide Semiconductor), which is an element that goes into building integrated circuits.

Fetch Cycle The first cycle in the fetch-decode-execute sequence of instruction execution. During the fetch cycle, the contents of the program counter are placed on the address bus, a READ signal is generated, and the instruction pointer is incremented. The data bytes arriving from the memory will be gated into the instruction register of the control unit. Some machines, including the Model 100, have a "fetch ahead" strategy that retrieves upcoming instructions before they are needed during slack memory cycles.

Fetch-Ahead See Look Ahead.

FF Flip-Flop or Form Feed. A control character used with printers which advances the paper so the top of the next sheet of paper, or the next form, is in position for printing.

FFFA See Low Byte.

FFFB See High Byte.

FFFF The hexadecimal representation of the maximum simple address value on 8-bit microcomputers: FFFF base 16 = 65535 base 10 = 177777 base 8 = 1111111111111111 base 2.

FFT Fast Fourier Transform. A fast computational algorithm for determining the Fourier curve fit to a collection of data points.

Field • File Extension—.BA

Field A logical grouping of data. It could be a group of related characters in a record (name field), a work area in memory, or, in the CPU, a zone within an instruction, such as opcode, address, or comment.

FIFO ("fife-oh") First-In-First-Out structure. Data is added at one end and removed from the other, like cars lined up at a gas station. A FIFO buffer is used to connect two devices operating asynchronously at different speeds. Each device is connected to one end of the FIFO buffer.

File A logical grouping of information given an identifying name and considered as a unit by a user. A file may be divided into records, blocks, or other units, as required by the memory device.

File, Data A file which contains data to be processed by a program. This is a matter of function rather than contents. For example, if you write a BASIC source program called SAMPLE.BA and then submit it to the BASIC compiler to produce a fast object program, the BASIC compiler will treat SAMPLE.BA as its input data rather than as a program to be executed.

File, Edit See Edit—Change Contents of a File.

File, Erase To erase any RAM file in BASIC, ASCII, or Machine language you need to Enter BASIC mode. While in BASIC key:

KILL "<filename>"

<filename> is the name of the RAM file as listed on the screen by the BASIC command files. You must include the file extension, .BA, .DO, or .CO, and enclose the entire file name in double quotes. Entering KILL "<filename>" erases the file <filename> from RAM.

File, Fragmented See Fragmentation.

File, Length of BASIC. Model 100 BASIC provides no means for determining the length of a file.

File, Machine Language The RAM file extension .CO designates the file as a Machine language file. The .CO extension is automatically assigned by the operating system when you use the LOADM command.

File, Object A file containing the 80C85 Machine language version of a program. In most cases the object file is a translation of a source program stored in a text file. Its extension is .CO. See Programs, Object.

File, Print To print the contents of the TEXT or BASIC Edit file you are in, hook up a printer, turn it on, and press the SHIFT and PRINT keys at the same

time. This produces the prompt "width," followed by the currently defined print width. The default width value is forty characters-per-line. Once you change the width value, it remains as you last specified until you change it again. If you are satisfied by the value given by the "width" prompt, press ENTER to begin printing. Otherwise, backspace and enter the width you want. Valid line width values range from 10 to 132. To interrupt printing before it is finished, press the SHIFT and BREAK/PAUSE keys at the same time.

In BASIC, pressing the SHIFT and PRINT keys together is equivalent to entering LLIST, which lists the program currently in BASIC memory. The print width will automatically be eighty characters-per-line.

File, Program A program file contains some type of program instructions specifying how data is to be processed. If you write a BASIC program called SAMPLE.BA this is a source program. If you then submit it to BASIC compiler to produce a fast object program, the BASIC compiler will treat SAMPLE.BA as its input data rather than as a program to be executed. The BASIC compiler will produce an object file with extensions .OBJ which contain a translation of your source program into the 8088 Machine language.

File, Text A file containing character data, letters, numbers, or special characters (.DO). These files may have any file extension. By contrast, an object file (.CO) contains data which is not displayable as characters. Most data and programs that you write will be text files. See Data Files.

File Control Block An area of memory used by a disk operating system to keep track of a file's status. It is used to keep track of input, output, or update, current records, and physical sector numbers.

File Directory, Cassette See CLOAD.

File Directory, RAM The table of contents which allows convenient access to specific files. The Main Menu contains a directory of all RAM files and built-in applications programs. To display a directory of RAM files while in BASIC, enter:

FILES

To get a directory of cassette files, use CLOAD with a file name you know is not on the cassette. See SPACE, CLOAD.

File Extension—.BA The RAM file extension signifying that the file contents are in tokenized BASIC format. The Model 100 operating system automatically assigns this extension to the file when it is saved to RAM.

File Extension—.CO The RAM file extension which

File Extension—.CO • File Management

designates the file as a Machine language file. The .CO extension is automatically assigned by the operating system when you use the LOADM command. See File, Object.

File Extension—.DO The RAM file extension which designates the file as a text document. The Model 100 operating system automatically assigns this extension to the file name.

File Handling Software See COPY+, DATA+.

File I/O Commands and Functions BASIC.
Key

I = input to BASIC only

O = output from BASIC only

I/O = input and/or output

= references the device by using an associated file number. Device, file number, and file name or configuration are associated by the OPEN statement and terminated by the CLOSE statement

@ = default device, redundant to specify this device

with this statement or command

[] = not I/O but file related (verify, delete, rename)

File Kinds and Extensions The Model 100 handles three types of type formats: ASCII, BASIC, or Machine language. RAM files have file extensions assigned by the operating system when the file is saved to RAM.

.DO an ASCII format document file.

.BA a tokenized BASIC file.

.CO a Machine language file.

Input, output files via the RS-232-C and modem interfaces are always in ASCII formats, as are files displayed on the LCD and printer. See File I/O, SAVE, SAVEM, LOAD, LOADM, and BASIC Editing.

File Management File management or diskette maintenance is record-keeping on diskettes. This includes creating them, finding them by name,

BASIC File I/O Command and Functions

BASIC Command or Function	RAM:	CAS:	COM:	MDM:	LCD:	LPT:
CLOAD		@ I				
CLOAD?		@ []				
CLOADM		@ I				
CLOSE	# I/O	# I/O	# I/O	# I/O	# O	# O
CSAVE		@ O				
CSAVEM		@ O				
INPUT #	# I	# I	# I	# I		
KILL	@ []					
LINEINPUT#	# I	# I	# I	# I		
LOAD	@ I	I	I	I		
LOADM	@ I	I				
MERGE	@ I	I	I	I		
NAME	@ I	[]				
OPEN	# I/O	# I/O	# I/O	# O	# O	
PRINT#	# I/O	# O	# O	# O	# O	
PRINT#USING	# O	# O	# O	# O	# O	
RUN	@ I	I	I	I	I	
RUNM	@ I	I				
SAVE	@ O	O	O	O	O	
SAVEM	@ O					

File Management System • File Types By Device

ensuring that adequate free space is available on the diskette, maintaining backups, and deleting files no longer needed. These functions are supported by the various DOS functions, but require thoughtful planning by the user to ensure proper results.

Some Database Management Systems (DBMSs) attempt to automate part of the work of keeping track of files and diskettes. Some DBM systems maintain files of control and tracking data on other files and diskettes, and may provide alternatives to DOS functions.

File Management System A group of programs designed to format and manage user files in a transparent way. The system allows symbolic names and attributes, and manages the physical allocation of storage. Usually part of the Operating System. See TEXT mode.

File Name See File Name and Extension.

File Name and Extension A file name is one-six characters in length. It may be preceded by an optional device name (separated by a colon). Also, an optional two character file extension may follow (separated by a period). The entire specification for a file consists of the three parts <device>:<file-name>.<extension> and is often called a <file-spec> in documentation. Example: RAM:BOOKS.DO Normally only RAM files will have extensions which are automatically assigned by the operating system when they are saved to RAM. File extensions are .DO (an ASCII file), .BA (an encoded BASIC file), and .CO (a Machine language file). Devices on the Model 100 are:

CAS:Cassette tape stored file
COM:RS-232-C transmission file
LCD:LCD screen transmission file
LPT:Printer transmission file
MDM:Modem transmission file
RAM:RAM memory stored file

For example of how BASIC uses device names in files see: LOAD, LOADM, MERGE, OPEN, RUN, RUNM, SAVE, and SAVEM, and File I/O.

File Separator A special pattern of bits or frequency which separates one file from another on tape or a disk.

File Size, RAM BASIC provides no means for determining file size. File names and sizes are listed in the segment of the memory map that constitutes the system menu. It stores the name and size of all the RAM files you see on the Main Menu. A BASIC program which will read this data from the memory

map and display it on the screen in tabular form is listed below:

```
10 REM File Directory with file lengths
20 LNE = 1
30 PRINT "[file] [size] free";TAB(23)
40 FOR ETRY = 0 TO 154 STEP 11
50 LSB = 63931
60 MSB = 63932
70 FLNM = 63933
80 FOR STP = 0 TO 5
90 X$ = CHR$(PEEK(FLNM+STP+ETRY))
92 IF X$ = CHR$(0) THEN X$ = " "
94 PRINT X$
100 NEXT STP
110 PRINT ".";
120 FOR STP = 6 TO 7
130 X$ = THEN CHR$(PEEK(FLNM+STP+ETRY))
132 IF X$ = CHR$(0) THEN X$ = " "
134 PRINT X$
140 NEXT STP
150 PRINT (PEEK(LSB+ETRY)+
  PEEK(MSB+ETRY)*256);
160 IF LNE = 1 AND ETRY < 143
  THEN PRINT "":LNE=0 ELSE
  LNE=1:GOSUB 200:PRINT TAB(23);
170 NEXT ETRY
180 IF INKEY$ = "" THEN 180
190 END
200 IF ETRY = 11 THEN PRINT " space";
210 IF ETRY = 44 THEN PRINT " is";
220 IF ETRY = 55 THEN PRINT (FRE(0)+256);
230 RETURN
```

The screen output from the program looks like this:

[file]	[size]	free	DUMP	.BA	32769
TEST1	.BA	32871	space	RED	.DO 34383
JELL	.DO	34306	is	MOM	.DO 33985
NUT	.DO	33978	27831	SIZE08	.DO 33967
SIZ257	.DO	33699		SPACE	.BA 32885
	.	0		.	0
	.	0		.	0
	.	0		.	0

File Specification The complete specification of a file, including file name, device identifier, and extension.

File Specifiers By Device See Device.

File Specifiers For Each Device See File I/O Commands and Functions.

File Types See File I/O Commands and Functions.

File Types By Device See Device.

FILES BASIC Command. Displays the names of all RAM files on the screen. This makes a RAM file directory available without leaving BASIC. Normally, you would go to the Main Menu to see a directory of RAM files. The format is:

FILES

Files, Merge ASCII with BASIC See MERGE.

Files in RAM, Delete To erase any BASIC, ASCII, or Machine language file, enter the BASIC command:

KILL "filename"

where filename is the name of the RAM file as listed on the Main Menu or on the screen by the BASIC command FILES. Include the file extension, .BA, .DO, or .CO and enclose the entire file name in double quotes. Entering KILL "filename" erases the file from RAM memory.

Files in RAM, List The Main Menu contains a directory of all RAM files and built-in applications programs. To display a directory of RAM files, enter the BASIC command:

FILES

To get a directory of cassette files, enter the BASIC command:

CLOAD "filename"

where "filename" is the name of a file you know is not on the cassette. See SPACE, CLOAD.

Files on Cassette, List See CLOAD.

Files Used by BASIC See MAXFILES.

Filespec A complete DOS file name, including the drive, file name, and extension. File specification, file spec, or filespec is used to show where the file specification goes in a typical command format. A DOS filespec has three parts:

<device>:<filename>.<extension>

A sample command is:

RAM:FILE2.DO

<device> is always optional.

<extension> may be omitted in some situations where the context provides a clear default value, such as .BA for BASIC programs and .DO for documents.

Filing Systems for Diskettes* Library cases for diskettes, which hold up to 100 diskettes and filing enclosures for tapes and magazines. C.R.C. Wholesale.

Financial Analysis Packages 1* Two versions of the same program for use with 8K and 16K Model 100's. This is a menu-driven program (with 20 different screens) that uses the function keys for menu selection. You can use it to determine the

original loan balance, the loan interest rate, the payment amount, the compound interest beginning and ending balances, equivalent compound interest rates, the ending balance and deposit amount required for annuity, the bond yield, and the bond purchase amount required for a desired yield. All input fields are verified. York Software.

FIND Command Key. F1 is the Find key in TEXT mode and can be used to define text from the cursor position to a specified word or number. It can also be used to locate a specified character or group of characters. See Phone Number, FIND a; String, FIND a; ADDRSS Mode; and INSTR.

Firmware A program stored in ROM. Originally, firmware was used only for microprograms inside the CPU. In microprocessors, many kinds of programs are in ROM, and firmware designates any ROM-implemented program. Firmware in the Model 100 includes the applications programs TEXT, ADDRSS, SCHEDL, TELCOM, and BASIC.

FIX BASIC Function. Returns the integer value of digits to the left of the decimal point and removes all digits to the right of the decimal point. The format is:

<variable> = FIX(<numex>)

<numex> may be any numeric expression

For example:

FIX(34.34)

returns 34, and

FIX(-76.89)

returns -76.

FIX is different from INT because it strips the decimal portion of your number, while INT returns the whole number not greater than numeric expression.

Fixed Media The physical devices on which information is recorded. The main media for the Model 100 are cassette and RAM. Media are classified as:

a) Removable Media—such as diskettes, cassettes, and some hard disks or

b) Fixed Media—such as most hard disks. Fixed media are not removable from the device that drives them, so there is no ability to store additional data or backup copies off-line (outside the computing system) for insertion when needed.

Fixed-Head Disk A disk system with a head over each track. This costly, but quick, system eliminates head positioning delay time.

Fixed-Point Integer representation with the decimal point assumed to be in a fixed position. See Floating-Point.

Flag A status indicator for a special condition. A flag is normally stored in either a flip-flop or a register. A microprocessor usually provides the following status flags: carry, zero, sign, overflow, and half-carry or auxiliary carry.

Flip-Flop A circuit used to store one bit of information. An FF is bistable with two stable states (0 and 1). Registers are generally assembled out of flip-flops.

Flippy Another name for a mini-floppy. Also used to describe a single-sided diskette which has been turned (flipped) over to record data on the other side.

Floating Gate A technique used for UV-erasable EPROMs. A silicon gate is isolated inside the silicon dioxide.

Floating-Point Representation of numbers in a fixed-length format, such as 24 or 32 bits. The number $\langle n \rangle$ is normalized and encoded as a mantissa field $\langle m \rangle$ and an exponent field. The name reflects the representation, which remains fixed as the decimal point floats, or as changes in magnitude are reflected by adjustment of the exponent field with renormalization of the mantissa field. The precision of the representation is limited by the number of bits allocated to the mantissa field. See Fixed-Point.

Floating-Point Package A set of software routines or hardware features necessary to perform the floating-point arithmetic functions of addition, subtraction, multiplication, and normalization. To assure precision, the design of an FPP requires careful analysis of error-propagation phenomena.

Floppy Disk A flexible mylar disk (diskette) for mass storage of information. The diskette is sealed in a square plastic jacket, lined with a soft material which cleans the diskette as it rotates. A cut-out slot provides access for the moving head which must actually come in contact with the diskette surface in order to read or write. Other holes in the jacket provide access to sector index holes in the diskette. Diskettes are hard-sectored if the sector start points are marked by holes in the diskette. Soft-sectored diskettes have only one (or a few) holes to mark the start of the track. The sector start marks are placed on the soft-sectored diskette, under software control, in a process known as formatting the diskette. Disks are also classified as single- or double-sided, double-, dual-, or quadruple (quad) density.

Floppy Mini A smaller floppy that is $5\frac{1}{4}$ " square compared to 8" for the original floppy.

Flowchart A symbolic representation of a process. Boxes represent commands or computations; diamonds represent tests and decisions (branches). It is useful to use a flowchart before writing your program because it helps you understand and debug the program by segmenting it into logical, sequential steps.

Flyback The time delay while the spot on a CRT comes back to the beginning of a screen.

FMS File Management System.

FOR...NEXT BASIC Statements. FOR and NEXT perform a series of instructions in a loop a given number of times. The format is:

```
FOR <countvar> = <startval> to <endval>
[STEP <increment>]
.
.
.
NEXT <countvar>
```

$\langle \text{countvar} \rangle$ is an integer or single-precision variable to be used as a counter.

$\langle \text{startval} \rangle$ is a numeric expression which is the initial value of the counter.

$\langle \text{endval} \rangle$ is a numeric expression which is the final value of the counter.

$\langle \text{increment} \rangle$ is a numeric expression to be used for incrementing or decrementing the counter. If omitted, BASIC assumes that the increment is one.

Within a single FOR...NEXT loop, all lines following the FOR statement are executed until the NEXT statement is encountered. Then, the counter $\langle \text{countvar} \rangle$ is increased or decreased by the amount specified by the STEP value $\langle \text{increment} \rangle$. If you don't specify an $\langle \text{increment} \rangle$, it is set to 1. A check is done to see if the value of the counter is now greater than the final value $\langle \text{endval} \rangle$. If it isn't greater, BASIC goes back to the statement directly after the FOR statement, and repeats the process. If it is greater, the statement following the NEXT statement is executed.

If the $\langle \text{increment} \rangle$ is negative, the check is reversed. The counter is decremented each time the loop is executed and execution continues until the counter $\langle \text{countvar} \rangle$ is less than the final value $\langle \text{endval} \rangle$.

The statements within the FOR...NEXT loop are executed once if the loop is encountered for the first time and $\langle \text{startval} \rangle$ is already greater than $\langle \text{endval} \rangle$ when the STEP value is positive; or b) if $\langle \text{startval} \rangle$ is less than the $\langle \text{endval} \rangle$ when the STEP value is negative. If $\langle \text{increment} \rangle$ is zero, an infinite loop will occur unless you provide some way to

set the counter greater than the final value.

Loops are nested when one or more FOR...NEXT loops are placed inside a larger FOR...NEXT loop. Whenever FOR...NEXT loops are nested, each loop must have a unique variable name. The NEXT statement for the inner loop must appear in a statement before the NEXT statement for the outside loop. However, if nested loops have the same end point, a single NEXT statement may be used for all of them.

Using the format:

NEXT <countvar1>,<countvar2>,<countvar3>
is the same as

```
NEXT <countvar1>
NEXT <countvar2>
NEXT <countvar3>
```

The variable or variables in the NEXT statement may be left out. In this case, NEXT returns to the most recent FOR statement. However, when using FOR ...NEXT loops, you should include <countvar> or <countvars> on all the NEXT statements. You should use variable names on all NEXT statements to avoid confusion. Using <countvar> on the NEXT statement will slow program execution.

When a NEXT statement is encountered before its corresponding FOR statement, the message "?NF error in <xxxx>" will appear on the LCD where <xxxx> is the line number of the unmatched FOR statement.

Some examples are:

```
OK
10 M=2:K=10
20 FOR I=M TO K STEP 2
30 PRINT I;
40 NEXT I
RUN 2
4
6
8
10
OK
```

Foreground Program The higher-priority program in multi-program execution. Also, a program that communicates a user or a process. See Background Program.

Form Feed To advance a page to the top of an EPSON RX-80 printer, enter the BASIC statement:

LPRINT CHR\$(12);

or use the "top of form" or "form feed" manual control button FF on the printer. You may then need to adjust the paper in the printer so that the perforations are at the top of the page.

If you are writing a program, you may want to provide instructions for a pause to allow the adjustment of the paper.

Formatter A circuit or program which writes file, track, and address marks, adds pre-ambls and post-ambls, and checks characters on disks or tapes.

Formulator A development system for the F-8 microprocessor.

FORTH A programming language and operating system. FORTH is characterized by threaded code and postfix, or reverse Polish notation. You can create new commands in FORTH in terms of the existing commands or in Machine language code. Your new commands then replace the standard FORTH language. This feature allows you to tailor-make your commands if you are designing a language to suit a particular application or problem. FORTH is also more transportable between microcomputers than BASIC and most other languages. FORTH is more like a Machine code than a high level language, and, although it may be somewhat harder to master than BASIC, it also runs much faster.

FORTRAN FORMula TRANslator. One of the first high-level languages, FORTRAN is still widely used, especially by scientists and engineers. The differences between BASIC and FORTRAN are slight, mostly dealing with I/O statements. It is rich in mathematical functions and supports an extended precision calculations mode for scientific problems.

FORTRAN is a compiled language, not an interpreted language like BASIC. A program is edited in a file, then submitted to a compiler for translation into executable object code. FORTRAN and BASIC are so similar that BASIC could be called an interpreted dialect of FORTRAN. Here is a sample FORTRAN program to sum the first 100 integers:

```
20 FOR I = 1 TO 100
30 J = J + I
40 NEXT I
50 END
```

Fourier Transform The mathematical analysis of a complex wave-form into harmonic components.

FPGA Field Programmable Gate Array. A device containing an array of AND and OR gates which can be re-connected (programmed) in the field (i.e. without return to factory).

FPLA Field Programmable Logic Array. A logic array which can be programmed. FPLAs are used to implement the control section of bit-slice processors.

FPLS Field Programmable Logic Sequence.

FPP Floating-Point Package.

Fragmentation A situation in which memory has been divided in such a way that it has many fragments of unallocated space that is too small to be useful or to allow optimum performance. Compacting all of the allocated areas into a single area opens up a large block of free space.

Frame The necessary underlying structure for a record, file, or other data item. The frame creates an organization into which the data is put.

Frankenstein Adventure* Find the hidden clues to awaken Frankenstein, while avoiding numerous pitfalls. 24K; cassette. SilverWare.

FRE BASIC Function. Returns the number of numeric memory bytes that are not being used by BASIC when <dumex> is numeric. When <dumex> is string, FRE displays the amount of unused string memory space. If you have executed a CLEAR statement without specifying the string space variable, FRE returns the default of 256 bytes of string space. The format is:

<variable> = FRE(<dumex>)

<dumex> is a string or numeric dummy argument. Since strings in BASIC can have variable lengths, they are manipulated dynamically. Each time you do something to a string, its length may change. For this reason, string space may become fragmented. FRE with any string value will collect all useful data and compress it, freeing unused areas of memory, and then return the number of free bytes. This is called housecleaning, and is done automatically when work space is needed. Periodic FRE(" ") executions mean shorter delays in clearing disk space. See CLEAR.

Freeze, Screen To freeze the screen briefly during program execution, write a delay loop:

```
1000 FOR Y = 1 TO 2000
1010 NEXT Y
```

To freeze the screen until the operator is done with it, put in a dummy input statement and instruct the operator to press ENTER to proceed. The input variable <A\$> is unnecessary. The format is:

```
1000 INPUT "Press ENTER to continue";A$
```

Freeze System Operation The BREAK/PAUSE key stops system operation. Pressing it twice reactivates the operation. See Control Keys for related information.

Frequency Number of cycles per second. $F=1/T$ where T is the period in seconds over which cycles are counted.

Front Panel A panel with lights and switches to display information and allow direct control or access to memory or registers, which requires a specific interface and a monitor program. For microcomputers with no front panel, all operations are performed from a keyboard and screen.

Front-End Processor A processor which interfaces with a user or a process. The front-end may perform pre-processing translations or file handling, while the main processor performs interpretation, execution, or other processing.

Frozen Keyboard, Restart If the keyboard "freezes up" and refuses to respond, you can restart the system and return to the Main Menu by pressing the RESET button on the rear panel of the computer.

FS File Separator.

FSC Full Scale range.

FSK Frequency Shift Keying. A 0 is represented by one frequency; a 1 is given a different frequency. These two tones are then transmitted over telephone or radio links and converted back to digital signals upon reception. See Modem.

Full Duplex A communication technique which allows data to be transmitted and received simultaneously. See TELCOM, and Duplexing.

Fully Decoded Selection A method of selecting memory locations or input/output devices through a full n-bit address (typically n=16). This requires the use of decoders, but allows full utilization of 64K possible addresses.

Function Key Each of the five built-in operating modes on the Model 100 defines the function keys differently. In some cases, the same built-in applications program will contain more than one definition for the same key. To list the function key definitions on the screen, press the LABEL command key. Press it twice to erase the display. In BASIC, you may change the function key definitions using the KEY statement. Function key definitions in each mode are found on the next pages.

() parenthesis indicate a temporary value established by the F1 key.

Functions Choose from Menu. See Menu.

Functions, Math BASIC. See Math Functions Table.

Functions, String BASIC. See String Functions Table.

Fundamental The base or carrier signal on which a data signal will be superimposed. It is usually a pure sine wave with no distortion.

F/V Frequency to Voltage Converter.

BASIC		
Function Key	Label	Function Performed
F1	Files	Lists the names of all the RAM files on the LCD screen.
F2	Load	Types LOAD" on the LCD. You specify what device and file to load from, then ENTER the line to BASIC.
F3	Save	Types SAVE" on the LCD. You specify what device and file name to save to, then ENTER the line to BASIC.
F4	Run	Enters RUN to BASIC. This runs the file currently in BASIC memory.
F5	List	Enters LIST to BASIC. This lists the current contents of BASIC memory on the LCD.
F6	---	Not used.
F7	---	Not used.
F8	Menu	Enters MENU to BASIC. This exits BASIC and returns you to the Main Menu.
Text		
Function Key	Label	Function Performed
F1	Find	Finds the first occurrence of a string within the current text file.
F2	Load	Loads the cassette file you specify into RAM as an ASCII format file. Leaves you in the file and in TEXT mode when the LOAD is complete.
F3	Save	Saves the current text file, in ASCII format, to cassette using the file name you specified.
F4	---	Not used.
F5	Copy	Copies a marked text block into the paste buffer. The original text remains unchanged.
F6	Cut	Copies a marked text block into the paste buffer and deletes the marked block from the text file.
F7	Sel	Marks or selects the first character of a text block. The spaces between this first text block character and where you subsequently place the cursor constitutes a marked text block which you may then Copy or Cut.
F8	Menu	Exits from TEXT mode and returns to the Main Menu.
TELCOM Entry Mode		
Function Key	Label	Function Performed
F1	Find	Finds and retrieves a phone number from the ADRS.DO file displaying it on the LCD. It temporarily changes the meaning of F3 and F4.
(F3)	(More)	Temporary value after pressing F1. Displays more phone numbers for the same name.
(F4)	(Quit)	Temporary value after pressing F1. Quits the Find initiated by F1.

Function Key

TELCOM Entry Mode		
Function Key	Label	Function Performed
F2	Call	Calls the phone number currently on the LCD. Works only when the computer is connected to a telephone using the direct connect modem cable.
F3	Stat	Lets you change the status of the communications parameters used by the modem or RS-232-C devices.
F4	Term	Press to enter TELCOM Terminal mode, which is used for direct computer to computer communications.
F5	---	Not used.
F6	---	Not used.
F7	---	Not used.
F8	Menu	Exits TELCOM Entry mode and returns to the Main Menu.
TELCOM Terminal Mode		
Function Key	Label	Function Performed
F1	Prev	Displays the previous eight lines of incoming computer to computer communications data.
(F3)	(More)	Temporary value after pressing F1. Displays the next screenful of occurrences of the string you are finding.
(F4)	(Quit)	Temporary value after pressing F1. Quits the Find initiated by F1.
F2	---	Not used.
F3	---	Not used.
F4	---	Not used.
F5	Lfind	Finds and prints on an attached printer all the occurrences in the ADRS.DO file of the string you enter.
F6	---	Not used.
F7	---	Not used.
F8	Menu	Exits ADDRSS mode and returns to the Main Menu.
SCHEDL		
Function Key	Label	Function Performed
F1	Find	Finds and displays on the LCD the first screenful of occurrences in the NOTE.DO file of the string you enter. Temporarily changes the definition of F3 and F4.
(F3)	(More)	Temporary value after pressing F1. Displays on the LCD the next screenful of occurrences in the NOTE.DO file of the string you entered.
(F4)	(Quit)	Temporary value after pressing F1. Quits the find initiated by F1.
F2	---	Not used.
F3	---	Not used.

Function Key

SCHEDL		
Function Key	Label	Function Performed
F4	---	Not used.
F5	Lfind	Finds and prints on an attached printer all the occurrences in the NOTE.DO file of the string you enter.
F6	---	Not used.
F7	---	Not used.
F8	Menu	Exits from SCHEDL and returns to the Main Menu.



G Codes. ASCII 71, HEX 47. g—ASCII 103, HEX 67.

G Ground. Also, Generate signal. The output from an adder connected to a carry look-ahead circuit that requires a propagate signal.

Gain The output-to-input amplification ratio.

Games See Alexis Adventures; Blockade; Frankenstein Adventure; and Reversi.

Games #2* Four games for the Model 100: "Checkers," to play against the computer or another person; "Williamsberg Adventure," in which you search for the golden horseshoe; "Maximum," in which you compete with an opponent or the computer for possession of the higher point value squares; and "Amazing," a maze game with optional sound effects in which you try to outrun two opponents. Silverware.

Gap The space between two records or two blocks of information on a cassette tape or disk. A gap is usually set to a predetermined value, such as all 1s. It allows blocks to be rewritten in either a slightly expanded or a reduced format, due to speed variations of the drive.

Garbage Collection A technique for collecting unavailable, unused space in a mass memory and making it available for reuse by any of several schemes.

Gate A single logic function. The NAND, NOR, AND, OR, XOR, and NOT functions are examples of gates.

Gauss A unit of flux density (1 Maxwell per square cm.), named after German mathematician, Karl F. Gauss.

GCR Group Coded Recording.

GE General Electric. Also, Greater than or Equal to, represented by \geq or \Rightarrow .

Generate Some computers generate BASIC line numbers automatically. The Model 100 BASIC does not perform this function.

GI General Instruments, a manufacturer.

Gibson Mix A statistically balanced mix of instructions that is representative of general data processing applications. It is one of many variations used for bench-mark testing.

Glitch A pulse or burst of noise. A small pulse of noise is called a snivitz. The word glitch is usually reserved for the more dangerous types of noise pulses which cause crashes and failures.

Global Variable A variable whose name and value are accessible throughout the program or application system. Contrast with a local variable, accessible only within the block where it is defined.

GND Ground.

GOSUB and RETURN BASIC Statements. GOSUB and RETURN are used together to branch to and return from subroutines. The format is:

GOSUB<linnum> . . <linnum>' begin subroutine.

RETURN

<linnum> is the line number of the first line of the subroutine.

With Gosub you can call a subroutine any number of times in a program or from within another subroutine.

The RETURN statement causes BASIC to branch back to the statement following the GOSUB statement that called the subroutine. A subroutine may contain more than one RETURN statement in case you want to return from different points in the subroutine. You may place subroutines anywhere in a program.

In order to keep your program from entering a subroutine that should not be executed, detour around it using a STOP, END, or GOTO statement. Refer also to ON...GOSUB ON...GOTO. BASIC Statements and BASIC Interrupt Commands.

GOTO BASIC Statement. Used to exit from the numerical program sequence to a specified line number. The format is:

GOTO<linnum>

<linnum> is the line number in the program you want to execute next.

GP • Graphics

The <linnum> specified after GOTO will be executed if it is an executable statement. If <linnum> is a non-executable statement, such as DATA or REM, then the program will continue execution until it comes to a line that can be executed.

GOTO can be used for two kinds of branching, conditional and unconditional.

An unconditional GOTO automatically branches program execution to the line number specified. A conditional GOTO branches to a given line number or bypasses the GOTO depending upon the outcome of some conditional test such as an IF...THEN...ELSE or ON ERROR construction.

In debugging, the GOTO statement can be used in direct mode to re-enter a program at a desired point after a STOP or BREAK/PAUSE and SHIFT key combination are used to break program execution. It works like a CONT statement to preserve all variable values currently in memory. A RUN statement, on the other hand, would clear all memory first. GOTO differs from CONT in that CONT continues program execution at the line where the break occurred while GOTO lets you resume execution at any line number you specify.

Refer also to ON...GOSUB, ON...GOTO, and ON ERROR GOTO.

GP General Purpose.

GPIB General Purpose Interface Bus. The IEEE 488-1975 interface bus standard. Also called ANSI Standard MC 1.1-1975, or IEC Bus, in Europe.

GP1024 Color Printer* This color ink-jet hard-copy device combines color, text, and graphics in a single unit. By utilizing silent ink-jet technology, it supplies complete, eight-color hardcopy with 125 shades. The device's resolution exceeds most commercially available graphics CRTs. With 120 dots-per-inch, the GP1024 prints up to 1440 dots in the horizontal, and 85 dots-per-inch in the vertical. Ink for 500 typical graphic images is supplied by a disposable snap-in cartridge. The carriage of the GP1024 has a rated life greater than 6000 hours MTBF. The color head has a proven reliability of more than 19 billion operations. Printacolor.

GRAPH Key The GRAPH key is located in the lower left segment of the Model 100 keyboard, directly to the left of the space bar. By pressing the GRAPH key and another standard key at the same time, you can generate forty special graphics characters. These are 8-bit ASCII codes 123-126, 128-159, 163, 176, and 180. If you press the SHIFT, GRPH, and another standard keyboard key at the same time, this generates another 32 special graphics

characters. These are 8-bit ASCII codes 224-255. See ASCII Code and Character Table.

Graph Key See ASCII, ASCII Character Codes Table.

GRAPH+* When used with the E+ files, the user is able to represent data in pie, bar, or line charts. A maximum of eighteen numbers and labels are used in formatting either columns or rows. The user is prompted with several questions relating to the type of chart, printer, dimensions, and file wanted. Percentages and values are printed along with the graphs. This is part of BusinessPak+. 16K; cassette. Portable Computer Support Group.

Graphic Packages See GRAPH+.

Graphics BASIC. Line and Box. LINE is used to draw a line or box on the screen. The format is:

```
LINE [(<xcoord1>,<ycoord1>)]  
-<xcoord2>,<ycoord2>) [,switch] [,B[F]]
```

<xcoord1>,<ycoord1> and <xcoord2>,<ycoord2> are the starting and ending coordinate locations that define the line (or the diagonal of the box) to be drawn. Each pair of coordinates represent a pixel. The screen is composed of 240 by 64 pixels because there is a zero pixel; horizontal <xcoord> values may be between 0 and 239, and vertical values <ycoord> may be between 0 and 63. The first pair of coordinates (<xcoord1>,<ycoord1>) may be omitted. If so, the starting point of the current line will be 0 if it is the first line drawn. If not, the starting point of the current line will be the second set of coordinates (<xcoord2>,<ycoord2>) used by the previous line statement.

<switch> is an optional numeric expression evaluating to an even or an odd number. If an even value is used, the current line is erased; if an odd value is used, the current line is drawn. If <switch> is omitted, BASIC assumes you want to set or draw the line.

B is optional. If included, it causes a box rather than a line to be drawn using the coordinate pairs given. To use B you must specify a <switch> value.

F is optional. If included, it fills the box drawn by the B option. To use F you must specify B and the <switch> value.

Graphics BASIC. Used to turn on the pixel at a specified location on the LCD screen. The format is:

```
PSET(<xcoord>,<ycoord>)
```

<xcoord> and <ycoord> are the x and y coordinates of the pixel to be turned on. Values for the x coordinate can range between 0 and 239. Values for

the y coordinate can range between 0 and 63. If you entered:

```
PSET(0,0)
```

the upper leftmost pixel would be turned on.

PRESET is used to turn off the pixel at a specified location on the LCD screen. The format is:

```
PRESET (<xcoord>,<ycoord>)
```

<xcoord> and <ycoord> are the x and y coordinates of the pixel to be turned off. Values for the x coordinate can range between 0 and 239. Values for the y coordinate can range between 0 and 63. If you entered:

```
PRESET (0,0)
```

the upper leftmost pixel would be turned off.

Graphics, String For animation effects, one of the best and simplest methods is string graphics using BASIC. To create string graphics, define each on-screen row of a graphics figure as a string value and then print the strings at changing locations.

The first step is to map out the graphics figures you are planning to animate on a graphics worksheet, such as the one in the back of your Model 100 manual. These graphics may be composed of alphanumeric characters and special graphics characters. To include one of the 8-bit graphics characters in a BASIC string use the CHR\$ statement with the ASCII code of the character you want as the argument. For instance, you may construct a simple boat figure using six block graphics characters on two lines. Each line must be defined as a separate string. For example, the boat figure could be represented by the following two strings:

```
A$=" "+CHR$(252)+CHR$(254)
B$=CHR$(253)+CHR$(239)+
CHR$(239)+CHR$(251)
```

To move the boat across the screen, print the two strings on adjacent rows at a sequence of consecutive screen locations, erasing the old image as you print each new one. Screen character locations range from 0 (the upper left corner of the screen) to 319 (the lower right corner of the screen). A FOR ...NEXT loop, using the loop counter to increment the screen location, is an excellent way of setting this up. For example:

```
05 REM loop to print moving figure
10 FOR p=40 TO 60
20 REM print the new image
30 PRINT @ p, A$
40 PRINT @ p+40, B$
50 REM erase the old image that was not
   printed over
```

```
60 PRINT @ p -1," "
70 PRINT @ p+39," "
80 NEXT p
```

This program segment prints the previously defined strings, A\$ and B\$, which together constitute the graphics figure, at a sequence of consecutive column locations across the screen. Screen positions 160 and 200 are in the same column on the fourth and fifth rows of the LCD respectively. Notice that the program prints blanks at the end of each loop to erase the segment of the old image that is not covered by the new image when it is printed. If this were not done, the new and old images would overlap and collect on the screen. The following program is an example of using string graphics for animation.

```
10 'Short Voyage Sequence
20 CLS
30 'Define the boat in a vertical orientation.
40 A$=" "+CHR$(252)+CHR$(254)
50 B$=CHR$(253)+CHR$(239)+
   CHR$(239)+CHR$(251)
60 'Define the ocean surface
70 WV$=STRING$(40, CHR$(131))
80 'Loop to sail the boat over the ocean
   surface
90 FOR P=40 TO 60
100 'Print the Ocean
110 PRINT @ 120, WV$
120 'Erase last loop image that will not be
   printed over by this loop image
130 PRINT @ P-1," "
140 PRINT @ P+39," "
150 'Print the boat image
160 PRINT @ P, A$
170 PRINT @ P+40, B$
180 , If short of contact point advance the
   projectile towards the boat
190 IF P<59 THEN PRINT @ 119-P, CHR$(143):
   PRINT @ 120-P," "
200 'Delay loop to slow the action
210 FOR H=0 TO 30: NEXT H
220 'End the boat sailing loop
230 NEXT P
240 'Define the boat in vertical orientation
250 C$=CHR$(252)
260 D$=CHR$(239)+CHR$(254)
270 E$=CHR$(239)+CHR$(251)
280 F$=CHR$(253)
290 'Erase segment of horizontal boat that
   won't be covered by vertical boat 300 PRINT
   @ 60," ":PRINT @ 100," "
310 'Loop to sink the boat
320 FOR D=100 TO 180 STEP 40
```

```

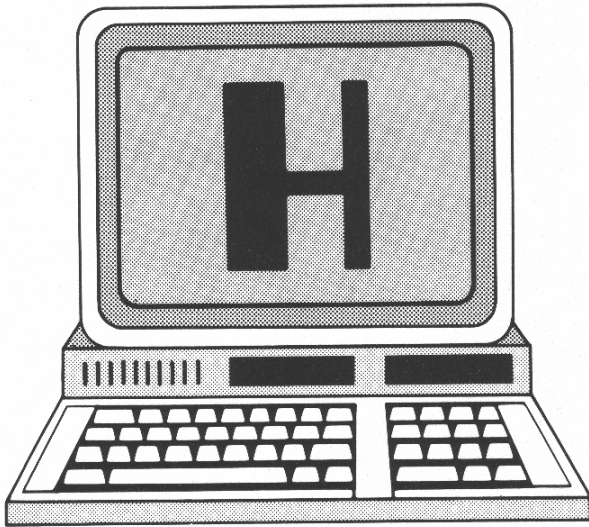
330 'Erase old vertical boat image and reprint
the ocean when boat under it
340 IF D-40 <> 140, THEN PRINT@ D-40,
" " ELSE PRINT@ D-40, CHR$(131)CHR$(131)
350 'Print the boat image, erase old boat image
not printed over this time
360 PRINT @ D+1, " "
370 PRINT @ D, C$
380 PRINT @ D+40, D$
390 PRINT @ D+80, E$
400 PRINT @ D+120, F$
410 'Delay loop to slow the action
420 FOR W=1 TO 50: NEXT W
430 'End loop to sink the boat
440 NEXT D
450 'Erase vertical boat that won't be covered
by new horizontal boat
460 PRINT@ 140, " ":PRINT@ 180, " ":PRINT
@ 221, " "
470 'Print the horizontal boat one more time
480 PRINT @ 218, A$: PRINT @ 258@, B$
490 'Loop to repeat rising bubble
500 FOR L=1 TO 7
510 'Loop to make one bubble rise
520 FOR B=218 TO 98 STEP -40
530 'Print one bubble
540 PRINT @ B, "O"
550 'If not the first loop erase the last bubble
image printed
560 IF B ≠ 218 THEN PRINT @ 98, CHR$(131)
570 'End loop to make one bubble rise
580 NEXT B
590 'End loop to repeat rising bubble
600 NEXT L
610 'Freeze the screen to prevent scrolling
and ok prompt
620 FOR Z=1 TO 10000: NEXT Z

```

Note that you may omit all the comment lines (prefaced by an apostrophe, which is shorthand for REM) to greatly reduce the program length.

Ground The point of reference in an electrical circuit (not necessarily the physical ground). The ground point is considered at nominal zero potential, and all other potentials in the circuit are compared with it.

GT Greater Than (also represented by >).



H Codes. ASCII 72, HEX 48. h—ASCII 104, HEX 68.

H Hexadecimal. Used as a suffix to denote hexadecimal numbers in Intel format. Also, High—the most significant half of a register or a pointer, usually bits 0 to 7 of a 16-bit word. Used in naming the high part of the 8088 registers—AH, BH, etc.

Half-Carry The carry from bit 3 into bit 4, which is required for correct addition of packed Binary Coded Decimal numbers (where two BCD digits reside in one 8-bit byte.)

Half-Duplex A mode of communication in which data may be transmitted in only one direction. See TELCOM.

Halt When a computer stops all activity.

Halt System Operation In most applications pressing the BREAK/PAUSE key once pauses system operation. Pressing the BREAK/PAUSE key twice continues operation. If this doesn't work, you can abort the current procedure by pressing the SHIFT and BREAK/PAUSE keys at the same time, but you usually can't restart an operation halted this way.

Hamming Code A 7-bit error-correcting code named after the inventor.

Handler A program used to control or communicate with an external device, such as a diskette drive.

Handshaking A basic communications synchronizing technique using two signals: 1—ready ? ; 2—yes/no acknowledgment. The handshaking procedure is carried out prior to any data transfer when establishing a connection between two data communication devices. For example, a CPU will ask an

I/O: is input buffer 1 empty? If yes, it can be reloaded. If no, the CPU must wait.

Hard Copy Computer output printed on paper. You can always print a copy of the current screen contents by pressing the PRINT key. To print the entire contents of an open file, press the SHIFT and PRINT keys at the same time.

Hard Disk A disk composed of a magnetic coating applied to a rigid substrate, such as aluminum or ceramic. The term is used to distinguish from "soft" (floppy) disks which are flexible. See Floppy Disk.

Hard-Sectored A disk in which the recording surface has been divided into sectors using non-alterable methods, such as a ring of holes in the disk itself. Hard-sectored disks require more hardware but can also store more information. See Soft-Sectoring.

Hardware Accessories See Bi-Tech Bar Code Reader; 8K Memory Models.

Hardware vs. Software Computer programs are called software. Software is usually used to program the hardware—the actual chips, wires, boards, etc., which make up the computer. A special case is Read-Only Memory (ROM), which is hardware that contains a permanent copy of software. These terms can be confusing; for instance, a "BASIC ROM" means a ROM (hardware) containing a copy of a BASIC interpreter program (software). Such ROMs are often called firmware to distinguish them from non-program hardware and from software in changeable media (RAM, diskette, cassette, etc.).

Harmonic An integer Multiple of a fundamental frequency.

Harmonic Distortion Distortion due to the signal's non-linear characteristics, resulting in output which includes harmonics of a harmonic-free sine input.

Hayashi, Suzuki One of the developers of the TRS-80 Model 100, whose name is permanently stored in ROM. The following program prints out ROM address locations 63901-63981. Draw your own conclusions.

```
5 FOR I=63901 TO 63981
10 LPRINT I,CHR$(PEEK(I)),PEEK(I)
15 NEXT
```

High Memory See CLEAR.

HIMEM Returns the highest address number available to BASIC. The value of HIMEM may be set

HLT • Human Engineering

using the CLEAR command. The default value for HIMEM is MAXRAM. For example:

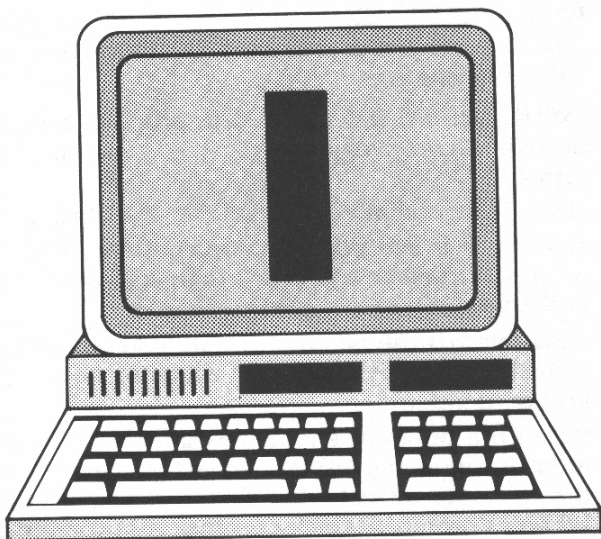
```
OK
PRINT HIMEM
62960
OK
```

HLT 80C85 Assembly Language Instruction. Halt. The processor is stopped and registers and flags are unaffected.

Holding Register A register that holds data temporarily to bridge a speed or timing gap between two devices.

Human Engineering If you write programs others will use, you should give clear prompts for every item of input data, freeze the screen long enough for information to be read, provide clear error messages if anything is entered incorrectly or error situations arise, etc. This is referred to as "human engineering"—making the program easy to use, as well as technically correct. Another term for this side of programming is "user friendly."

Several guidelines exist for writing user friendly programs. If a complex series of data items has been entered and some entries turn out to be invalid, the user should be able to reenter the bad items without redoing everything. Error messages should indicate not only that an entry is invalid, but also just how it is invalid and, if possible, give hints on correcting it.



I Codes. ASCII 73, HEX 49. i—ASCII 105, HEX 69.

I DEBUG Command. Inputs and displays in hexadecimal form, one byte from the specified port address. CS is the default register. The format is:

I (hex port address)

IC Integrated Circuit.

IF...THEN...ELSE BASIC Statement. IF statement tests a condition for true or false and directs further processing based on the results of the test. The formats are:

IF <expression> THEN <clause> [ELSE <clause>]

or

IF <expression> GOTO <line> [ELSE <clause>]
<expression> any relational or logical expression which can be evaluated to 0 (true) or not zero (false).

<clause> A sequence of BASIC statements separated by colons, or the number of a line to branch to. If either the first or the second clause is a GOTO statement, then BASIC accepts a THEN lnum or ELSE lnum as an implied GOTO.

<line> the line number of a line that exists in the program.

If the value of <expression> is not zero (true), the THEN <clause> or GOTO <line> is executed. If the <expression> is equal to zero (false) the THEN <clause> or first GOTO <line> is skipped and the ELSE <clause> is executed instead.

An ELSE clause is not required. In this case if the <statement> yields false, execution continues with the next executable statement.

When you enter an IF...THEN statement in direct mode, if you haven't entered a line with the speci-

fied number, it directs control to a line number, a "?UL ERROR", or undefined line number message is displayed.

When using IF to test equality for a value that is the result of a single- or double-precision computation, remember that the internal representation of the value may not be exact. Single- and double-precision numbers are stored internally in floating-point, binary format, so the test should be against the range over which the accuracy of the value may vary.

IF...THEN...ELSE statements may be nested. Each statement must contain the same number of THEN and ELSE clauses; otherwise, each ELSE is matched with the closest unmatched THEN.

Immediate Response to One-Character Answers BASIC. Use INKEY\$ to provide immediate response to a one-character answer (Y or N, etc.). The program can process the request without you having to hit ENTER after the response. For example:

```
10 PRINT "Make another copy Y/N?"
20 ANS$ = INKEY$: IF INKEY$ = " " THEN
  GOTO 20
30 IF ANS$ = "Y" OR ANS$ = "y" THEN GOTO
  1000
40 IF ANS$ = "N" OR ANS$ = "n" THEN GOTO
  2000
50 GOTO 10
```

This technique can be used with a numeric response by using the VAL function:

```
20 ANS$ = INKEY$: IF INKEY$ = " " THEN
  GOTO 20
30 NUMBER = VAL(ANS$)
```

IN 80C85 Assembly Language Instruction. Input. The data placed on the eight-bit bi-directional data bus by the specified port is moved to register A. The addressing mode is direct. No flags are set.

Indexed Sequential Access Method A program or package that supports files organized with one or more indexes. Records are retrieved from the file either sequentially or randomly depending on the key used in the index.

Information Services See CompuServe; Dow Jones News/Retrieval; Knowledge Index; and Source.

Infoworld Computer Oriented Newspaper. *Infoworld* is a weekly magazine about microcomputers. Valuable for keeping up with new developments on a weekly basis. Most computer stores and larger bookstores carry *Infoworld* and a good assortment of other relevant maga-

Initial Program Load • INPUT#

zines. *Infoworld's* address is:

InfoWorld
Circulation Dept.
Box 837
Framingham, MA 91701

Initial Program Load Normally, when you first turn on the Model 100, the Main Menu is the initial program load and is automatically displayed and waits for you to indicate which applications program or RAM file you wish to use.

First, enter BASIC from the Main Menu and write or LOAD the program you want to auto-run. Then, SAVE the program to RAM, and enter:

IPL "<filenm>"

where <filenm> is the name you gave to the BASIC file to autorun, and turn off the computer. The next time you turn the computer on, that program will automatically execute.

INKEY\$ BASIC Variable. Reads a single character from the keyboard. The format is:

<stringvariable> = INKEY\$

This assigns a keyboard character to the <stringvariable>.

INKEY\$ reads only one character, regardless of the number of characters waiting in the keyboard buffer. BASIC checks the keyboard only once; if no key is pressed, it assigns a null character, " ", to the string variable. You can pause execution until input is received from the keyboard by following the INKEY\$ assignment with a conditional test. This test returns execution to the line with the INKEY\$ assignment program line if the string variable holds a null value, " ". For example:

```
10 KEY$ = INKEY$
20 IF KEY$ = " " THEN GOTO 10
```

In this case, the GOTO in GOTO 10 may be omitted because the conditional looping occurs anyway.

The result of INKEY\$ must be assigned to a string variable before using the character with any BASIC statement or function. When INKEY\$ is in use, no characters are displayed on the screen.

If you press ENTER in response to INKEY\$, the carriage return character passes through to the program.

INP BASIC Function. Returns the byte read from CPU port <n>. It is the complementary function to the OUT statement. See OUT. The format is:

<variable> = INP(<n>)

<n> the port number to be read within the range 0

to 255.

INP is synonymous with the IN instruction in Assembly language.

INPUT BASIC Statement. Halts program execution while it receives input from the keyboard. The format is:

INPUT["<prompt>"]<variable>[,<variable>...]

<prompt> is an optional string constant used to prompt the person using the program to enter the desired input.

<variable> is the name of the numeric or string variable(s) or array element(s) which receives the keyboard input.

When the program encounters an INPUT statement, it pauses and displays a question mark (?) on the screen to indicate that data is to be entered. When a <prompt> is included, the prompt string is displayed before the question mark. When an INPUT statement includes more than one variable, these variables are either entered on one line or entered separately followed by returns. If BASIC receives a return but is expecting more variables, it prompts you with two question marks (??).

The data entered is stored in the variable(s) declared in the variable list. Each data item must be separated by commas or a return, and the total number and type of items should match the number and type of the variables in the list. Strings entered in response to an INPUT statement don't need quotation marks unless they contain commas or significant leading or trailing blanks. Numbers may be input into string variables but are saved in ASCII format.

If you respond to INPUT with too many data items, an "?Extra ignored" message appears on the screen. If the wrong type of value is entered (string data instead of numeric data), BASIC will display a "?Redo from start" error message. If a single variable is requested and you don't want to enter any value, you can press ENTER to get the default values of 0 for numeric input or null for string input. No input values are assigned to any variables until an acceptable response is given.

INPUT See Internal Memory.

INPUT Statement See Freeze Screen Display.

INPUT# BASIC Statement. Reads data items sequentially from a device or file and assigns them to program variables. The format is:

INPUT# <filenum>,<variable>[,<variable>,...]

<filenum> is the number used when the file was opened for input.

<variable> is the name of a variable that has an

item in the file assigned to it. It can be a string or numeric variable, or an array element. Variables should be separated by commas.

The sequential file may be located in RAM, on cassette, or it may be a sequential data stream from the RS-232C port or the modem. Data in the input file should be separated by commas, and the type of data in the file must agree with the type specified by the variable name. No question mark is displayed in the screen with INPUT# (unlike the INPUT statement).

The data items should appear as if the data were entered as responses to an INPUT statement. When using numeric values, the leading spaces, carriage returns, and line feeds are ignored. The first character encountered which is not a space, carriage return, or line feed is assumed to be the start of the number. The number ends with a space, carriage return, line feed, or comma. BASIC automatically performs any conversions necessary between numeric variable types as it inputs numeric data into corresponding numeric variables.

If BASIC is scanning for a string item, the leading spaces, carriage returns, and line feeds are ignored. The first character found that is not a space, carriage return, or line feed is assumed to be start of the string item. If this first character is a quotation mark ("), the string item consists of all characters read between the first quotation mark and the next quotation mark. If the first character of the string is not a quotation mark, the string is an unquoted string; it will end with a comma, carriage return, or line feed, or after 255 characters have been read. If the end of the file is reached when a numeric or string item is being input, the item is cancelled.

INPUT\$ BASIC Function. INPUT\$ returns a string of <n> characters, read from the keyboard or from file number <filenum>. The format is:

```
<stringvariable> = INPUT$(<n>[,#<filenum>])
```

<n> is the number of characters to be read. It may be any numeric value in the range 1 to 255.

<filenum> is the optional file number used on the OPEN statement. <n> number of characters are read from the file opened as <filenum> to the <stringvariable> in the INPUT\$ statement.

When <filenum> is omitted, the keyboard is used for input. No prompt is displayed and input characters are not echoed on the screen. INPUT\$ accepts all characters except BREAK.

Input/Output BASIC. See BASIC Input/Output Commands by Device.

Input/Output See Input, and Output.

Input/Output BASIC Data See BASIC File I/O Commands and Functions.

INR 80C85 Assembly Language Instruction. INcrement Register. The content of <r> is incremented by one. All condition flags except CY are affected. The addressing mode is register. Z, S, P, and AC flags are set.

INR M 80C85 Assembly Language Instruction. INcrement Memory. The content of the memory location. The address is in the H and L registers, and is incremented by one. All condition flags except CY are affected. The addressing mode is register indirect. Z, S, P, and AC flags are set.

Insert Character BASIC. If you have not yet pressed ENTER to send the line to BASIC memory, backspace (which erases the characters you backspace over) until you reach the place you want to insert the character. Then, rekey the portion of the line you erased.

If you have already entered the line to memory, you have two options. You can rekey the entire line, using the same line number and inserting the character you left, and enter the new line to memory. Otherwise, enter EDIT and the number of the line with the missing character and press ENTER. This translates the line to TEXT mode where you may insert the character according to the TEXT editing procedure. Press F8 (Exit key) to return the line to BASIC memory in its edited form, where it replaces the old version of the same line.

Insert Characters TEXT. See EDIT.

Insert Line BASIC. To insert a line into a program, give the new line a number between the numbers of the lines you wish to insert it between. For example, to insert a line between program lines 30 and 40, give the new line the number 35.

INSTR BASIC Function. Searches for the first occurrence of one string within another and returns the position of the match. The optional offset <n> sets the position within the search string for starting the search. The format is:

```
<variable> = INSTR([<n>],  
  <searchstr>,<matchstr>)
```

<n> is an optional integer value in the range 1 to 255, which specifies the position within the search string where the search should begin. If the number is omitted, the search begins at the first position and searches the whole string.

<searchstr>,<matchstr> can be string variables, expressions, or constants. <searchstr> is the string

Instruction • Interrupt

to be searched and <matchstr> is the string to be matched.

If the start option <n> is greater than the length of <searchstr>, or if the search string is empty, or if the <matchstr> is not found, INSTR returns zero (0). If <matchstr> is empty, INSTR returns <n>, or one if <n> is not specified. When <n> is not between 1 and 255 the message "?FC ERROR", an illegal function call message, is displayed on the LCD.

Instruction Causes a computer to do a specific action. Commands differ from instructions in several ways. A command is usually a complete specification of an action, while it usually takes many more instructions to make a useful program; commands are usually executed immediately by the computer, while instructions are saved for later execution in a program; commands are executed by the operating system of the computer, while instructions must first be processed by a program.

INT BASIC Function. Returns the largest integer that is less than or equal to <x>. The format is:

<variable> = INT(<x>)

<x> is any numeric expression.

For example:

PRINT INT(57.97) returns 57.
PRINT INT(-4.53) returns -5.
PRINT INT(99.999) returns 99.

See FIX and CINT.

Integer Convert to integer by truncation in BASIC. See CINT.

Integer Variable BASIC. Integer, or numeric, variables for whole numbers from -32768 to +32767, end in % or start with a letter specified in a DEFINT statement.

Integer variable names must start with a letter and can have any number of characters although only the first two are significant. It must not be any reserved word, such as IF, ON, THEN, GOTO, etc., or a reserved word followed by a type declaration character (\$, %, !, #). See BASIC Reserved Words and BASIC Variable Names.

Integrated Circuit A complete electronic circuit with multiple components (transistors, diodes, resistors, capacitors, etc.) constructed on a single silicon chip.

Integrity of Data Insuring that data (or programs) cannot be altered improperly. For example, in a payroll system, steps must be taken to insure that employees cannot improperly alter their pay rates or hours worked. Data security consists of guaran-

teeing both data integrity and data secrecy or privacy.

Interface The point at which two systems make contact. Most microcomputers have multiple interfaces or "ports," such as serial, parallel, monitor, power, and joystick ports. Interface is also used to refer to the type of interconnection, depending on its size or shape (subminiature 25-pin D connector), its mode of function (serial, parallel, etc.), or its electrical characteristics (RS-232, IEEE, etc.). The Model 100 has a RS-232-C serial interface, a Centronics standard parallel printer interface, a Hewlett-Packard HEDS-3000 compatible bar code reader interface, two female audio DIN plug interfaces for modem and cassette cables, and a 40-pin external bus interface.

Internal Commands The command processor is a program which accepts a command (usually from a keyboard) and causes it to be carried out. Another type of command processor carries out some commands directly (internal commands), but also locates and runs other programs (external commands). Some command processors contain the programming required for all commands they process. Others do not carry out any commands directly, but examine them, determine what other program can carry it out, locate the program, and let the program take it from there. See BASIC Command Table.

Internal Memory Memory is any device which can store information and allow it to be retrieved when needed. The Model 100 relies primarily on Random Access Memory (RAM) and cassettes. Internal memory includes RAM, the general purpose, erasable and reusable memory located inside the computer, and ROM (read-only memory), which contains fixed data. This can be read and used by the computer but cannot be changed with a WRITE or OUTPUT command. Many operating systems are in ROM. See Bubble Memory, External Memory, Memory, Memory Map, and Memory Address.

Interpreter A program which translates a source program written in a high-level language, such as BASIC, into Machine language. The source program contains relatively readable statements, which are translated by a interpreter, producing an object program.

Interrupt A signal to a microprocessing unit that an event has occurred that requires attention. The unit saves enough information to resume the task it is currently working on, and then executes code from an interrupt servicing program. If there are

multiple interrupt lines, the MPU can determine what type of event has occurred (vectored interrupts), from the type of interrupt signal it receives. Interrupts provide a much more efficient way of managing external events than continually checking to see if any event has occurred (polling). Model 100 BASIC allows you to define subroutines for handling interrupts from the modem, RS-232C, function keys, and system clock.

Interrupt Commands, BASIC See BASIC Interrupt Commands.

INX 80C85 Assembly Language Instruction. Increment register pair. The content of the register pair <rp> is incremented by one. No condition flags are affected. The addressing mode is register.

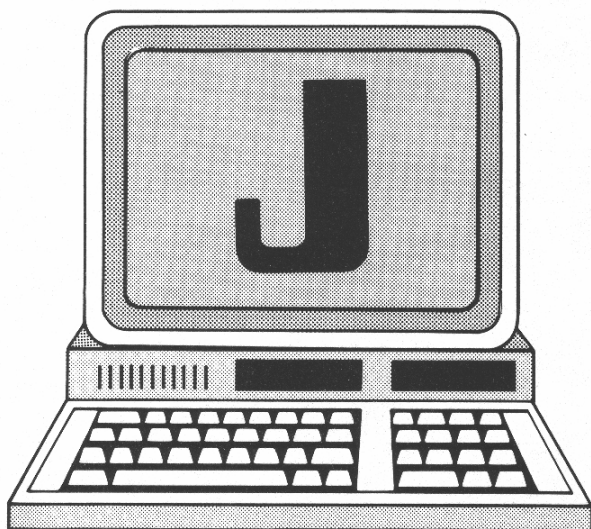
I/O See Input/Output.

IPL Initial Program Load. Starting the computer.

IPL BASIC. See Initial Program Load.

IR Instruction Register. In most MPUs, the IR contains the address of the next instruction to be executed. Branches are implemented by loading a new value into the IR.

ISAM See Indexed Sequential Access Method.

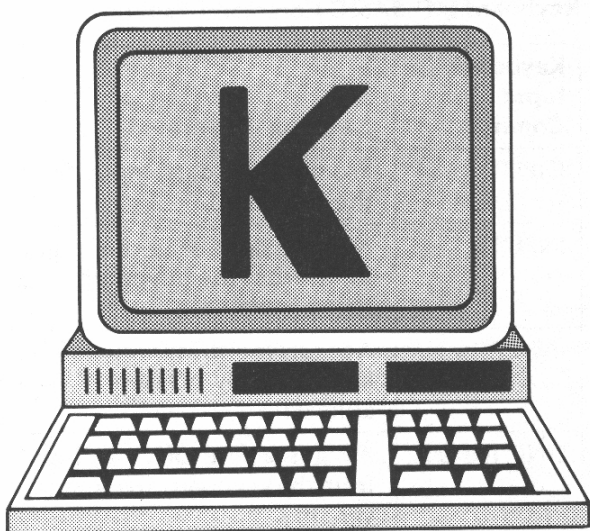


J Codes. ASCII 74, HEX 4A. j—ASCII 106, HEX 6A.

J 80C85 Assembly Language Instruction. Conditional Jump. If the specified condition is true, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise, control continues sequentially. The addressing mode is immediate. No flags are set.

JMP 80C85 Assembly Language Instruction. JuMP Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction. The addressing mode is immediate. No flags are set.

Justify To make the edges of a text file line up straight. Left justification is almost universal in English text. Right justification is usually present only in text that has been typeset or processed by a word processor or special typewriter to insert a variable amount of space between words or letters (proportional spacing). The Model 100 text editor left justifies the text it prints, no matter what line length you chose. It does not right justify the text.



K codes. ASCII 75, HEX 4B. k—ASCII 107, HEX 6B.

K Sign for 1000. Used to specify amount of storage in computer memory—usually KB, or 1000 bytes. See KB.

KB Kilobytes. Measurement of bytes in thousands.

KEY BASIC Statement. The KEY statement allows function keys to be defined as soft keys. When pressed and followed by a return, the specially defined key inputs any string (up to fifteen characters), into BASIC. If you include a carriage return (ASCII character) in the definition, the string is automatically entered.

KEY <num>,<x\$>

and

KEY LIST

<num> is the function key number in the range one to eight.

<x\$> is a string expression which will be assigned to the key. String constants must be enclosed in quotation marks.

If you have not altered the original function key definitions, the soft keys have the following values:

F1	FILES	F2	LOAD''
F3	SAVE''	F4	RUN
F5	LIST	F6	Not Used
F7	Not Used	F8	MENU

KEY LIST lists all eight soft key values on the screen. The fifteen characters of each value are displayed. If any key is undefined, no value will be listed. KEY LIST has the same effect as pressing the LABEL command key.

KEY <num>,<x\$> sets the value of <x\$> to the function key specified (one to eight). <x\$> can be one to fifteen characters long. If it is longer than fifteen characters, the first fifteen characters are the only ones assigned, the rest are truncated. The “?FC Error” an (Illegal function call error), is displayed when the value for <num> is not in the range one to eight. In this case, the previous key assignments are retired. If you omit <x\$>, the message “?MU Error”, the (Missing operand error) is displayed.

Assigning a null string (zero length) to a soft key disables the function key.

To reset the function keys to the values they had originally, use the following two command lines to call the two Machine language subroutines stored in ROM.

CALL 23164,0,23366

and

CALL 27795

See ON KEY GOSUB.

KEY ON/OFF/STOP BASIC Statement. KEY(n) ON/OFF/STOP is used in conjunction with the ON KEY statement to activate and deactivate trapping of a specified function key in a BASIC program. The format is:

KEY<n> ON

or

KEY<n> OFF

or

KEY<n> STOP

<n> is a numeric expression in the range one to eight. It indicates which of the function keys are to be trapped.

KEY<n> ON must be executed to activate trapping of function key activity. After KEY<n> ON, every time BASIC starts a new statement it checks to see if the specified key was pressed. If so, it performs a GOSUB to the first line of the subroutine specified in the associated ON KEY statement.

If KEY<n> is OFF, no trapping takes place, and even if the key is pressed, the event is not remembered.

When a KEY<n> STOP statement has been executed, no trapping takes place. However, if you press the specified key, your action is remembered so that an immediate trap takes place when KEY<n> ON is executed, transferring program execution to the first line of the subroutine designated in the associated ON KEY statement.

See ON KEY GOSUB.

Keyboard • Keys, Programmable Function

Keyboard The Model 100 keyboard is a full sized full action keyboard consisting of the standard qwerty keyboard and twenty-five other special keys. All keys automatically repeat if held down. The special keys include eight programmable function keys in two groups of four, left to right, above the highest row on the standard keyboard. Most programmable function keys are defined and used differently by each of the five built-in applications programs. They are all user-programmable in BASIC using the ON KEY statement. Current function key definitions in each application may be displayed on the eighth line of the LCD using the LABEL command key to toggle the display on and off. The remaining function keys on each row are, left to right: four command keys, Paste, Label, Print, and Break/Pause, and four cursor movement keys. These keys are used in same way in all applications. Other special keys are DEL/BKSP, ENTER, NUM, CODE, GRPH, CTRL, TAB, and ESC. For further discussion of each special key or set of keys see individual entries.

Keyboard—ECHO ECHO means to send characters typed on the keyboard to the screen for a visual confirmation of what has been typed. There is no hardwired connection between the keyboard and the screen. The keyboard simply enters characters into memory. The ROM programs of Model 100 then copy the characters from memory to the screen, creating a duplicate or "echo" of what was typed.

Keyboard, Read Data from BASIC. See INKEY\$, INPUT\$, LINE INPUT, ON KEY GOSUB.

Keyboard I/O BASIC Command Table

Keyboard Input Command	Function Performed
INPUT	Prompts for data input from the keyboard
INPUT\$	Assigns a string of a given length, input from the keyboard, to a string variable
INKEY\$	Accepts the string value of the key currently pressed as variable data
LINE INPUT	Assigns a line of data, input from the keyboard, to the string variable
ON KEY GOSUB	Defines an interrupt subroutine to execute when a specific function key is pressed.

Keys, Programmable Function Each of the five built-in operating modes on the Model 100 defines the function keys differently. In some cases, a given built-in applications program contains more than one definition for some keys. You may view the current function key definitions on the LCD, while in any application, by pressing the LABEL command key. Pressing the LABEL key a second time erases the LABEL display. In BASIC you may change the function key definitions using the KEY statement. Function key definitions in each mode are:

Programmable Function Keys		
Function Key	Label	Function Performed
F1	Files	Enters the FILES BASIC command to list the names of all the RAM files on the LCD screen.
F2	Load	Types LOAD" on the LCD. You specify what device and file to load from, then ENTER the line to BASIC.
F3	Save	Types SAVE" on the LCD. You specify what device and filename to save to, then ENTER the line to BASIC.
F4	Run	Enters RUN to BASIC. This runs the file currently in BASIC memory.
F5	List	Enters LIST to BASIC. This lists the current contents of BASIC memory on the LCD.
F6	---	Not used.
F7	---	Not used.

Keys, Programmable Function

Programmable Function Keys		
Function Key	Label	Function Performed
F8	Menu	Enters MENU to BASIC. This exits BASIC mode and returns you to the Main Menu.
		Text
Function Key	Label	Function Performed
F1	Find	Finds the first occurrence, within the current text file, of the string you input within the current text file.
F2	Load	Loads the cassette file you specify into RAM as an ASCII format file. The file is in TEXT mode when the LOAD is complete.
F3	Save	Saves the current text file, in ASCII format, to cassette using the file name you specified.
F4	---	Not used.
F5	Copy	Copies a marked text block into the paste buffer. The original text remains unchanged.
F6	Cut	Copies a marked text block into the paste buffer and deletes the marked block from the text file.
F7	Sel	Marks or selects the first character of a text block. The spaces between this first text block character and the cursor position constitutes a marked text block which you may then Copy or Cut.
F8	Menu	Exits from TEXT mode and returns to the Main Menu.
		TELCOM Entry Mode
Function Key	Label	Function Performed
F1	Find	Finds and retrieves a phone number from the ADRS.DO file and displays it on the LCD. It temporarily changes the function of F3 to F4.
(F3)	(More)	Temporary value after pressing F1. Displays more phone numbers for the same name.
(F4)	(Quit)	Temporary value after pressing F1. Quits the Find initiated by F1.
F2	Call	Calls the phone number currently on the LCD. Works only when the computer is connected to a telephone using the direct connect modem cable.
F3	Stat	Lets you change the status of the communications parameters used by the modem or RS-232-C devices.
F4	Term	Press to enter TELCOM Terminal mode, for direct computer to computer communications.
F5	---	Not used.
F6	---	Not used.

Keys, Programmable Function

Programmable Function Keys		
Function Key	Label	Function Performed
F7	—	Not used.
F8	Menu	Exits TELCOM Entry mode and returns to the Main Menu.
TELCOM Terminal Mode		
Function Key	Label	Function Performed
F1	Prev	Displays the previous eight lines of incoming computer to computer communications data.
(F3)	(More)	Temporary value after pressing F1. Displays the next screenful of occurrences of the sought for string.
(F4)	(Quit)	Temporary value after pressing F1. Quits the Find initiate.
F2	—	Not used.
F3	—	Not used.
F4	—	Not used.
F5	Lfind	Finds and prints all the occurrences in the ADRS.DO file of the entered string on an attached printer.
F6	—	Not used.
F7	—	Not used.
F8	Menu	Exits ADDRSS mode and returns to the Main Menu.
SCHEDL		
Function Key	Label	Function Performed
F1	Find	Finds and displays on the LCD the first screenful of occurrences in the NOTE.DO file of the string you enter. Temporarily changes the definition of F3 and F4.
(F3)	(More)	Temporary value after pressing F1. Displays on the LCD the next screenful of occurrences of the string you entered into the NOTE.DO file.
(F4)	(Quit)	Temporary value after pressing F1. Quits the find initiated by F1.
F2	—	Not used.
F3	—	Not used.
F4	—	Not used.
F5	Lfind	Finds and prints on an attached printer all the occurrences of the string you entered into the NOTE.DO file.
F6	—	Not used.
F7	—	Not used.
F8	Menu	Exits from SCHEDL and returns to the Main Menu.

() parentheses indicate a temporary value established by the F1 key.

Keyword A word which has special significance to a program. It must not be misspelled or used for other purposes, or erroneous results may occur. The BASIC keywords are reserved for use by BASIC only and may be used as variable names. See BASIC Reserved Words.

KILL BASIC Command. Used to delete a file from RAM memory. The format is:

KILL "<filespec>"

<filespec> is the name of the RAM file as it appears on the Main Menu. It must include the two-character file extension. You must be in BASIC to use KILL to delete BASIC, TEXT, and Machine language RAM files. It is the only way short of erasing the entire RAM memory at once.

Occasionally, when fewer than 200 bytes of free memory are left, BASIC will not delete the file named by the KILL command. In this case, you may open the file and delete lines or blocks until there is enough free memory to kill the remainder of the file. Or, you can temporarily store another file in the paste buffer, kill the target file, and paste the file in the paste buffer back into the RAM file it came from.

Knowledge Index* A new reference information service for owners of many models of personal computers. Knowledge Index lets you search over four million articles, reports, and books. There is also a comprehensive collection of over 10,000 indexed journals covering numerous topics.

Knowledge Index is made up of subject sections that cover popular information topics such as business, news, magazines, films, and various professions. Each section has one or more databases which consist of descriptions and abstracts of literature in the field covered.

The central storage of the databases is within the DIALOG Information Services facility in Palo Alto, California. When you dial, you are connected to one of their computers.

The hours that you can access Knowledge Index are Monday through Thursday from 6 p.m. to 5 a.m., Friday from 6 p.m. to midnight, Saturday from 8 a.m. to midnight, and Sunday 3 p.m. to 5 a.m. on Monday. You need a modem cable or a built-in modem and an acoustic coupler, and you must pay a one-time fee of \$35.00 that can be charged to any credit card and entitles you to a User's Handbook and two free hours of online practice time. The online cost of accessing the system is \$24.00 per hour and includes all network costs (Telenet and Tymnet.) Additional fees are charged for copies of articles and documents.

To receive special assistance, call 415-858-3796 between 7:00 p.m. and 11:00 p.m. Eastern Standard Time Monday through Friday. California residents call 800-227-5510.

To access Knowledge Index through either Telenet or Tymnet data communications networks, you must use the local access codes for the specific city you are calling from. To connect to the system, you need your User Number and your password.

The procedure for logging on via Tymnet is:

Call the local Tymnet number for your area
"System asks for terminal identifier" Type in the letter A after message stops. Type in terminal identifier.

Enter KI and press RETURN key.

System responds with "Host is online" and, after a short pause, "DIALOG Information Services Enter account number"

Enter six-character account number that begins with U.

Enter password.

System will give you instructions on how to use service and give the prompt ?

The procedure for logging on via Telenet is:

Call the local Telenet number for your area.

Tap the RETURN key twice and wait for response.

When system responds with "Telenet ... Terminal=", you enter D1.

When you receive the @ sign, enter C 41548K (be sure to leave a space after the C).

System responds "DIALOG Information Services Enter account number"

Enter your six-character account number beginning with U.

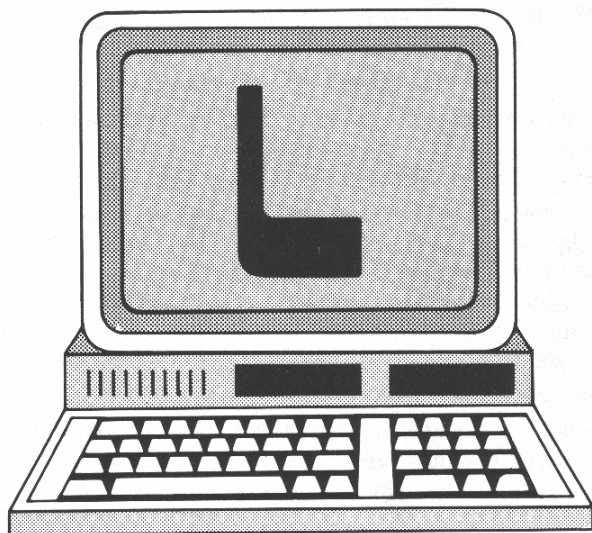
Enter password.

System will give you instructions on how to use it and give you the prompt ?

You may also connect to Knowledge Index via Direct Dial by calling 415-858-2738, entering your account number and your password.

Here is a brief list of sections currently available on Knowledge Index: Agriculture, Business Information, Computer and Electronics, Corporate News, Education, Engineering, Government Publications, Magazines, Medicine, News, and Psychology.

As a subscriber to Knowledge Index you receive a quarterly newsletter that helps keep you up-to-date on searching techniques and the databases available. This newsletter also serves as a way for users to communicate with each other on issues of interest. DIALOG Information Services, Inc.



L Codes. ASCII 76, HEX 4C. I—ASCII 108, HEX 6C.

LABEL Command Key. Used to turn on and off an LCD display showing the function key definitions in the current operation mode. The definitions are displayed on the eighth line of the LCD directly above a row of numbers permanently printed on its lower edge. Each of these numbers corresponds with the function key of the same number. In some built-in application programs such as TEXT and BASIC the labels disappear when you enter the program from the Main Menu; pressing LABEL brings them back. In the ADDRESS, SCHEDL, and TELCOM modes the label display turns on when you enter the program; pressing the LABEL key turns it off. See Key List.

Lap Warmer Games Series #1* A series of programs for play by one person, designed to use minimal memory and to load quickly. This four-game package comes on one cassette and includes two bonus games. The programs are: "Memory Challenge"—use your logical skills to deduce the number which the computer picks; "Repeater"—match memories with the computer; "Deflections"—find the hidden reflectors by firing rays into a grid and watching where they go; "Who Buys?"—let the computer decide who will pick up the tab; and "Bio-Lines"—use biorhythms to determine your good and bad days in advance. Solitary Software.

LCOPY BASIC Command. Causes a screen to dump to the system printer. Synonymous with the PRINT key. Data unable to be printed is ignored. The format is:

LCOPY

LCD Input Output Commands

LCD I/O Command	Function Performed
CLS	Turns off all LCD screen pixels
CSRLIN	Gives cursor's column position on the LCD
LCOPY	Prints current contents of the LCD to the printer
LINE	Draws a line or box on the LCD between coordinate points
POS	Gives cursor's line position on the LCD
PRESET	Turns off the LCD pixel at the screen coordinates given
PRINT	Prints data beginning at the current cursor position
PRINT@	Prints data at a specified LCD screen location
PRINT USING	Prints data on the LCD using a specified format
PSET	Turns on the LCD pixel at the screen coordinates given
SCREEN	Turns the function key label line (8) on and off
TAB	Tabs the cursor to the given LCD screen position

LDA 80C85 Assembly Language Instruction. Load Accumulator direct. Moving the content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, to register A. The addressing mode is direct. No flags are set.

LDAX 80C85 Assembly Language Instruction. Load Accumulator indirect. The content of the memory location, whose address is in the register pair, is moved to register A. Only register pairs <rp>= B (registers B and C) or <rp>= D (registers D and E) may be specified. The addressing mode is register indirect. No flags are set.

Leather Case* This hard leather case has an adjustable carrying strap that lets you carry it as a briefcase or a shoulder bag. Features include flaps that fold up to let you use the Model 100 keyboard and I/O ports while the computer is still in the case. Also included is a compartment for carrying accessories. Alpha 100.

LEFT\$ • LINE

LEFT\$ Returns the leftmost <n> characters of <x\$>. The format is:

<stringvariable> = LEFT\$(<x\$>,<n>)

<x\$> may be any string expression.

<n> is a numeric expression which must be in the range 0 to 255. It specifies the number of characters you want in the result.

If <n> is larger than LEN(<x\$>), the entire string <x\$> is returned. If <n> = 0 a string of zero length is returned.

See also LEN, MID\$ and RIGHT\$.

Left Arrow Moves the cursor left one character. Holding down the left arrow (←) key makes the cursor continue left. Sometimes, when the cursor reaches the leftmost position it wraps around to the rightmost position of the previous line and repeats the process until it reaches the top of the file. In other cases the cursor left movement is confined to the current line. In all modes except TEXT and BASIC EDIT backspacing the cursor with the ← key erases the characters the cursor passes over.

In TEXT and BASIC EDIT, the ← key performs additional functions if used with the SHIFT and CTRL keys. Pressing SHIFT and ← moves the cursor to the first character of the current word. If it is already there, the cursor moves to the first character of the next word to the left. Pressing CTRL and ← moves the cursor to the first character of the current line.

Left Cursor BASIC. Moves the cursor one space to the left with the character the cursor backspaces over being deleted by the cursor movement. See BASIC Editing.

Left Justify See Justify.

Leggs* A set of four clear, acrylic legs that are installed in existing holes in the Epson RX-80 and IBM printers, which raises the printer so that a 3-inch pad of paper can be slid underneath. Paper can be accessed from all four sides. Easy to assemble. Argus, Inc.

LEN BASIC Function. Returns the number of characters in a string expression. The format is:

<variable> = LEN(<stringex>)

<stringex> may be any string expression.

The count returned includes blanks and unprintable characters.

Length of File No means for determining the length of BASIC files is available.

Length of String BASIC. See LEN.

LET BASIC Statement. Assigns the value of an expression to a variable. The format is:

LET <variable> = <expression>

<variable> is the name of the variable or array element which is to receive a value. It may be a string or numeric variable or array element.

<expression> is the expression whose value is assigned to <variable>. The type of expression (string or numeric) must match the type of the variable (string or numeric), or the type mismatch error message, "TM Error," will be displayed. Numeric expressions are automatically transformed into the appropriate numeric variable type by BASIC.

In BASIC programs, you don't need to use LET when assigning values to variables. The format is:

<variable> = <expression>

LHLD 80C85 Assembly Language Instruction. Load H and L Direct. Moving the content of the memory location whose address is specified in byte 2 and byte 3 of the instruction, to register L. The content of the memory location at the succeeding address is moved to register H. The addressing mode is direct. No flags are set.

LINE BASIC Statement. Used to draw a line or box on the screen. The format is:

LINE [(<xcoord1>,<ycoord1>)] -
(<xcoord2>,<ycoord2>) [,switch] [,B[F]]

<xcoord1>,<ycoord1> and <xcoord2>,<ycoord2> are the starting and ending coordinate locations that define the line to be drawn. Each pair of coordinates represents a pixel. The screen is composed of 240 by 64 pixels because there is a zero pixel. Horizontal (xcoord) values can be between 0 and 239, and vertical values (ycoord) between 0 and 63. If the first pair of coordinates (<xcoord1>,<ycoord1>) is omitted, the starting point of the current line will be 0,0 if it is the first line drawn. If it is not the first line drawn, the starting point of the current line will be the second set of coordinates (<xcoord2>,<ycoord2>) used by the previous line statement.

<switch> is the optional odd or even numeric expression. If an even value is used, the current line is erased; if an odd value is used, the current line is drawn. If <switch> is omitted, BASIC assumes you want to set or draw the line.

B is optional, and if included, causes a box rather than a line to be drawn using the coordinate pairs given as a diagonal. To use B, specify a <switch> value.

F is optional and fills the box drawn by the B option. To use F, specify B and the <switch> value.

Line—Continue on Next Line of Screen • Lines Per Inch, Six

Line—Continue on Next Line of Screen BASIC. Continues the same logical line on the next screen line if you enter more than forty characters without a return.

Line, Blank LPRINT with no other specifications prints a blank line (that is, feeds the paper up one line and returns to left margin) so that you can format your printout neatly.

Line, End Current Pressing ENTER ends the current line, sends the line to the requesting program, and puts the cursor at the start of the next line. This is valid in all modes except TEXT and BASIC Edit, where pressing RETURN signals the end of a text block or paragraph, as well as contributing line feed. See Control Keys.

Line Feed To advance one line on the printer (space up) without a carriage return, enter BASIC statement:

LPRINT CHR\$(10)

or use the “line feed” button (LF) on the printer. Entering just LPRINT gives a line feed—both a space up one line (line feed) and a return to left margin (carriage return).

Line Graphics See LINE.

LINE INPUT BASIC Statement. Reads all characters (except trailing blanks) of a line of 254 characters and assigns it to a string variable. Pressing ENTER stops the string assignment. The format is:

LINE INPUT [<prompt>];<stringvar>

<prompt> is a string constant that is displayed on the screen before input is accepted. A question mark is not printed unless you include it with the prompt string.

<stringvar> is the name of the single-string variable or array element to which the line is assigned. All input except trailing blanks are assigned to <stringvar>.

Pressing SHIFT/BREAK/PAUSE lets you exit from LINE INPUT and return to command level. To continue at the LINE INPUT, enter CONT. The <prompt>, if any, is repeated and data can be entered.

LINE INPUT# BASIC Statement. Reads an entire line from a sequential file into a string variable, ignoring delimiters. It reads all characters except following blanks, into the <stringvar>. The format is:

LINE INPUT#<filenum>,<stringvar>

<filenum> is the number under which the file was opened.

<stringvar> is the name of a string variable or array element to which the line will be assigned.

Reads all characters in the sequential file up to a carriage return. The carriage return/line feed sequence is skipped over and the next LINE INPUT# reads characters up to the next carriage return. The line feed/carriage return characters are returned as part of the string. See Open.

Line Number BASIC. The largest possible line number for a BASIC program is 65529.

Line Number, Error On BASIC. See ERR and ERL.

Line Spacing See Epson Printer.

Line-Oriented Editing In all modes except TEXT and BASIC EDIT, editing is restricted to lines not yet entered to BASIC or other programs, such as ADDRESS or SCHEDL. See EDIT.

Lines—Screen Display See LIST.

Lines, Deleting BASIC Program To delete program line 100, enter:

100

and press RETURN.

To delete a large number of program lines, it is best to enter EDIT and the range of numbers you wish to delete. This transfers them to TEXT mode and you can use the text editing functions to do block deletions.

To delete a block of text in TEXT (synonymous with BASIC EDIT mode) place the cursor over the first character to delete and press F7, the Select key in TEXT. Move the cursor to the end of the lines to be deleted. The block appears in inverse video. Press F6, the Cut key, to delete the lines and F8, the Exit key, to return to BASIC. The lines are now far, far away, in the hazy land of deleted lines that few have ever seen. Are you satisfied?

Lines, Erasing Program To start a new program, enter:

NEW

SAVE before you do this if there is anything in BASIC memory that you want, as all lines are erased. If you don't erase the program in memory before starting on another, you will get an unusable combination of lines from your old and new programs.

Lines Per Inch, Eight To set this line spacing, enter the BASIC statement:

LPRINT CHR\$(27);“0”;

See Type Formats.

Lines Per Inch, Six To set this line spacing, enter the BASIC statement:

LPRINT CHR\$(27);“2”;

Lines Per Inch (7/72) • List RAM File Size

This is the default value when you turn the printer on. See Type Formats.

Lines Per Inch (7/72) To set the Epson RX-80 Printer to this line spacing, enter the BASIC statement:

```
LPRINT CHR$(27);"1";
```

This is a good setting for spacing with compressed print. See Type Formats.

Lines Per Page To set page length to <n> lines per page, enter the BASIC statement:

```
LPRINT CHR$(27);"C";CHR$(<n>)
```

<n> is between 1 and 127 and represents lines per page.

Link Time The point in the processing of a program with a language translator, Compiler or Assembler, when the program is tailored for a specific memory location. This occurs after compiling (compile time) but before execution (execution time). Some small systems do not require linking.

LISP A symbol-oriented programming language, with a simple and elegant syntax. It best handles objects whose structure cannot be completely specified in advance. LISP permits the building and discarding of intermediate structures without bothering the user with having to find the space necessary for storage. LISP encourages good programming style and facilitates modular programming.

LIST BASIC Command. Displays specified lines from the program currently in BASIC memory. The format is:

```
LIST [<line1>[-<line2>]]
```

<line1> and <line2> are valid line numbers in the range 0 to 65529. <line1> is the first line to be listed and <line2> is the last. A period (.) in place of either line number or alone indicates the current line number, which is the last line number edited, LISTed, or RUN.

You may use a dash and one line number to imply a range of lines to list, or two line numbers and a dash to specify a bounded line number range.

Three options are available when using the hyphen (-):

If you declare only

```
LIST <line1> -
```

that line and all higher numbered lines will be listed.

If you declare only

```
LIST - <line2>
```

all lines from the beginning of the program through <line2> are listed.

If you declare both line numbers

```
LIST <line1> - <line2>
```

all lines from <line1> through <line2>, inclusive, are listed.

Using LIST alone lists all line numbers in the program. Pressing F5, the List key in BASIC, is the same as entering LIST.

To freeze the list, press the BREAK/PAUSE key once. To resume listing after a pause press the BREAK/PAUSE key a second time.

List All Files on A Diskette See FILES.

List Cassette Files See CLOAD.

List RAM File Size BASIC provides no means of determining file size. File names and sizes are listed in the segment of the memory map that constitutes the system menu, which stores the name and size of the RAM files on the Main Menu. A BASIC program that reads and displays this data from the memory map is listed below.

```
10 REM File Directory with file lengths
20 LNE = 1
30 PRINT "[file][size] free";TAB(23)
40 FOR ETRY = 0 TO 154 STEP 11
50 LSB = 63931
60 MSB = 63932
70 FLNM = 63933
80 FOR STP = 0 TO 5
90 X$ = CHR$(PEEK(FLNM+STP+ETRY))
92 IF X$ = CHR$(0) THEN X$ = " "
94 PRINT X$
100 NEXT STP
110 PRINT ".";
120 FOR STP = 6 TO 7
130 X$ = CHR$(PEEK(FLNM+STP+ETRY))
132 IF X$ = CHR$(0) THEN X$ = " "
134 PRINT X$
140 NEXT STP
150 PRINT (PEEK(LSB+ETRY)+PEEK(MSB=
    ETRY)*256);
160 IF LNE=1 AND ETRY < 143 THEN PRINT
    " ":LNE=0 ELSE LNE=1:GOSUB 200:
    PRINT TAB(23);
170 NEXT ETRY
180 IF INKEY$ = " " THEN 180
190 END
200 IF ETRY = 11 THEN PRINT " space";
210 IF ETRY = 33 THEN PRINT " is";
220 IF ETRY = 55 THEN PRINT (FRE(0)+256);
230 RETURN
```

The screen output from the program looks like this:

[file]	[size]	free	DUMP	.BA	32769
TEST 1.BA	32871	space	RED	.DO	34383
JELL .DO	34306	is	MOM	.DO	33985

```

NUT .DO33978 27831 SIZE08 .DO33976
SIZ257.DO33699      SPACE .BA 32885
. 0                  . 0
. 0                  . 0
. 0                  . 0

```

List RAM Files The table of contents of a file system which allows convenient access to specific files. The Main Menu contains a directory of all RAM files and built-in applications programs. To display a directory of RAM files while in BASIC, enter:

FILES

To get a directory of cassette files, use CLOAD with a file name you know is not on the cassette.

LLIST BASIC Command. Displays on the printer all specified parts of the program currently in computer memory. This format is:

LLIST[<line1>]-[<line2>]

<line1> and <line2> refer to the range of line numbers to be listed.

LLIST can be interrupted by <SHIFT> <@>. Press any key to continue. BASIC always returns to command level after LLIST is executed.

LLIST works exactly like LIST, but output is to the printer.

LLIST 100-

lists line 100 to the end of the program to the line printer.

LLIST 100-200

lists line 100 through 200 to the line printer.

LLIST -100

lists all lines up to and including line 100 to the line printer.

See also BASIC Statements—Table of Formats, Descriptions.

LOAD BASIC Command. Takes a BASIC program from the specified device and places it into BASIC memory. It can also run the program automatically. The formats are:

LOAD ["<dev>:"] [<filename>"] [,R]

or

LOAD "<dev>:[<r>]<W> <P> <6> <S>" [,R]

<dev>: specifies the device from which the file is to be loaded. File data may be loaded into BASIC from the following devices:

RAM: is a file stored in RAM memory.

CAS: is a file stored on cassette tape.

COM: is a communications file received via the RS-232-C interface.

MDM: is a communications file received via the built-in modem.

If no device is specified, BASIC assumes the device is RAM:.

The RAM: and CAS: devices may be accessed for stored-file data using the first format which uses a <filename> to specify a stored file.

<filename> is a string expression naming the stored file. If the device is RAM: you must specify a <filename> and you can include the optional file extension shown for the file on the Main Menu. It is .DO for an ASCII format document file, .CO for a Machine language file, or .BA for a tokenized BASIC file. If the device is CAS:, no file extension is used and the <filename> is optional, also. If omitted, BASIC LOADs the first file encountered on the cassette tape.

The COM: and MDM: devices may be accessed for communications file data using the second format which specifies a list of communications parameters to be used in transmitting the communications file. Each communications parameter is specified by a single value in the format sequence <r> <w> <p> <s>.

<r> is the baud rate. Valid values range between 1 and 9, where each value stands for a different baud rate.

1=75 baud

2=110 baud

3=300 baud

4=600 baud

5=1200 baud

6=2400 baud

7=4800 baud

8=9600 baud

9=19200 baud

If the communications file to be loaded is specified as coming from the modem (MDM:) then the <r> communications parameter should be omitted. The modem will run at 300 baud by default. If this is altered, even by specifying an <r> value of 3 (300 baud, the built-in modem will be disabled.

<w> represents word length in bits and may be set to values 6, 7, or 8.

<p> represents parity. The default is E, meaning even parity and may be set to:

0 meaning odd

I meaning ignore

N meaning none

 indicates stop bits, 1 for one stop bit and 2 for two stop bits.

<s> enables and disables XON/XOFF status. Valid values are E which enables and D which disables it.

Load Binary Data with BASIC • Loop, Delay

LOAD removes all variables and program lines currently in computer memory and closes all open files. If (,R) is omitted, BASIC returns to direct mode after the program has been loaded. If the (,R) option is implemented with LOAD, the program is run after it is loaded.

When CAS: is specified as the device and you include the optional <filename>, BASIC searches the cassette tape until it finds the file with that name. Every time BASIC encounters a file which is not <filename> during the LOAD, it prints SKIP: and the name of the file skipped over on the LCD. To get a listing of all files on a cassette tape, rewind the tape and use

LOAD "CAS:<filename>"

where <filename> is definitely not on the tape. BASIC will then skip and list all the file names on the tape to the LCD.

Load Binary Data with BASIC (Machine language programs, etc.). See LOADM, CLOADM.

Load Module A file containing object code which is ready to be loaded into memory.

Load Time The point in the processing of a program when all translation and linking are completed and the program is loaded from disk or tape into memory for execution.

Loading Problems—Cassette Files See Cassette File Loading, Troubleshooting.

LOADM BASIC. Loads a Machine language RAM or cassette file to BASIC. The format is:

LOADM "[<dev:>][<file.nm>]"

<dev> the device that the file to be loaded is currently saved in. It may be a cassette (CAS:) or RAM (RAM:) file. If no device is specified then BASIC assumes the device is RAM by default.

<file.nm> is the name of the file that is to be loaded to BASIC. The extension (.nm), if used, should be .CO in the case of a RAM file, but it is usually not used at all.

Prior to loading a Machine language file, a segment of high memory is usually protected from use as BASIC workspace by using the CLEAR command. See CLEAR, BASIC, Address Notation, CLOADM.

LOCATE NEC PC-8201A BASIC Statement. Locate is used to position the cursor on the screen. Additional parameters can be used to change the cursor's size or make it blink. The format is:

LOCATE (row)<,(col)<,(cursor)<,(start)<,(stop)>

(row) is a number between 1 and 8 indicating the screen line number where the cursor is to be placed.

(col) is a number between 1 and 40 indicating the screen column number where the cursor is to be placed.

(cursor) indicates whether or not the cursor is visible. A 1 indicates visible, a 0 not visible.

(start) is the scan line start number. Must be a number between 1 and 31.

(stop) is the scan line stop number. Must be a number between 1 and 31.

The start and stop parameters are used to determine the size of the cursor. Cursor, start, and stop are not implemented in graphics mode.

When the parameters are omitted, the current values are assumed. Any values outside of the legal ranges cause an "Illegal function call" message.

Location of Cursor Find in BASIC. See CSRLIN and POS.

Locked-Up-Keyboard How to Restart. Press the reset button at the rear panel of the computer to unlock the keyboard, reset the system and return to the Main Menu.

LOG BASIC Function LOG function returns the natural logarithm of <x> (log to the base (e)). The format is:

<variable> = LOG(<x>)

<x> must be a numeric expression which is greater than zero.

Log of Screen Displays on Printer Pressing the PRINT key will print whatever is displayed on the screen on the printer, as it is displayed. This creates an ongoing printed log of everything you do.

Logical Operator BASIC Logical Generator Table (See table next page.)

Logon* This is electronic mail software designed to interface the Model 100 with mainframes such as those made by IBM and Honeywell. It allows you to access files, leave messages, and review them. Portable Computer Support Group.

Loop See FOR...NEXT.

Loop, Delay To freeze the screen briefly during program execution while the operator using your program reads a message, write a delay loop after you print the message:

1000 FOR Y = 1 TO 2000

1010 NEXT Y

To freeze the screen until the operator is done with it, put in a dummy input statement instructing the operator to press ENTER to proceed. The input variable (A\$) need not be used in your program:

1000 INPUT "Press ENTER to continue";A\$

Loop-Back Synonym for ECHO. Loop-back or ECHO is used to test the circuits of an input/output device by "looping" whatever is sent out back into the computer as if it were input. In this way the circuits inside the computer are tested in isolation from the circuits in the external device (and vice-versa) to help isolate a fault.

Lotus 1-2-3* A second generation spreadsheet and first generation spreadbase. As a spreadsheet it is huge: 2048 rows by 256 columns. It offers fifteen operators, forty-one functions, and sixty-six commands. Add to this the graphing and database functions, and the total number of commands is 110. Written in Assembly language, 1-2-3 has become the benchmark for calculation speed. With twenty-six macro keys, customized applications can be developed as well as single key-stroke commands. As a spreadbase, 1-2-3 provides an information management base. Creating and changing data is as easy as adding or subtracting field names. Data can be entered and extracted directly or by user-created forms. This information may be sorted on primary or secondary keys, retrieved by thirty-two criteria, have histograms produced on all or part of

the database, and be statistically manipulated using mean, count, standard deviation, and variance. This same information may be copied into a separate worksheet, manipulated, altered, and transported into a file of an entirely different application program such as dBASEII*, VisiCalc*, or WordStar*.

A major feature of 1-2-3 is the "typing alternative." This feature allows the user to select menu commands by simply moving the cursor keys to the desired command and pressing the ENTER key or typing the first letter of the command. Menu commands are logical extensions of 1st generation spreadsheet commands and are easily learned.

1-2-3 distinguishes between labels and numbers by alpha or value components. This allows the user to enter descriptive data and numerical data in a fashion similar to word processing.

With the range commands, groupings of cells can be named, moved, erased, combined into other worksheets, sorted, displayed in separated windows simultaneously, and edited to any degree. Ranges may be summed and presented in a variety of graph formats and changed

Logical Operator BASIC Logical Generator Table

Logical Operator	Description	First (fbit)	Second (sbit)	Result (rbit)
AND	If fbit and sbit are both 1, rbit is 1	1 0 1 0	1 1 0 0	1 0 0 0
OR	If either fbit or sbit is 1, rbit is 1	1 0 1 0	1 1 0 0	1 1 1 0
XOR	If either bit is 1 while the other bit is 0, the rbit is 1	1 0 1 0	1 1 0 0	0 1 1 0
EQV	If both bits have the same value, rbit is 1	1 0 1 0	1 1 0 0	1 0 0 1
IMP	If fbit is 1 and sbit is 0, rbit is 0, otherwise rbit is always 1	1 0 1 0	1 1 0 0	1 1 0 1
NOT	If fbit is 0, rbit is 1	1 0		0 1

Low Battery Indicator • LPRINT—Page Length

Using the forms capability, entire applications such as accounting systems, general ledgers, small databases, invoices, self-running tutorials, and demos can be implemented with no knowledge of programming. Furthermore, by incorporating the cell protect and unprotect feature, data may be safely entered by a computer novice without fear of overwriting important information.

Perhaps the true power of 1-2-3 lies in its ability to quickly move between files. One could open an active phone list recalling any number by first or last name, department, floor, color of eyes, shoe size, or any other criteria; then jump into a spreadsheet listing the price of gold on the NYSE, update the file with the latest prices; then go to a mailing list and print out labels based on selected criteria pertinent to the last file. This entire process can be done with less than ten keystrokes, none of which need be remembered because of the excellent menu descriptions shown in the control panel.

The Lotus 1-2-3 package contains a user's manual, a plastic slip-on function key template, warranty/registration card, quick reference booklet, and four diskettes: the 1-2-3 system disk; the 1-2-3 system disk backup; the graph program disk; and a 1-2-3 tutorial disk.

1-2-3 requires a minimum of 128K of memory, two double-sided drives, or one double-sided drive and a hard disk, and an RGB monitor or a monochrome display. Printers may be parallel or serial, and standard baud rates of 110 to 19200 are supported. Lotus 1-2-3 provides a valuable benchmark against which to measure the Model 100's software.

Low Battery Indicator When using four AA alkaline batteries as the power source for the Model 100, you can use the low battery indicator to tell when they need replacing. It is located on the upper right corner of the computer's face, right below the Radio Shack label. When the low battery light goes on, approximately twenty minutes of AA battery power remain. But don't worry—when the battery power is gone, the built-in ni-cad (nickel-cadmium battery) backs up the RAM memory, preserving your 32K RAM files for eight days and your 8K RAM files for thirty days. You cannot, however, turn off and use the computer until you replace the batteries or plug into wall current using the Radio Shack Power Supply unit (RS-26-3804). This recharges the ni-cad battery.

LPOS BASIC Function. Returns the current position of the print head within the printer buffer. The format is:

LPOS (<dummynum>)

<dummynum> is a numeric expression which is a dummy argument.

LPRINT BASIC Statement. Prints the values of the variables in its <expression list>, on the printer. The format is:

LPRINT<expression list>

<expression list> contains variables or constants, numeric or string, each separated from the others by a comma or a semicolon. String constants must be enclosed in quotes. The expression list may also include other BASIC commands such as CHR\$ and TAB.

When the items in the <expression list> are separated by commas, the printer advances to the beginning of the next print zone before printing the next value. A new print zone begins every fourteen spaces, or columns, starting with the zero column. If the expressions in the <expression list> are separated by semicolons, BASIC does not add any extra spaces to the values of the <expression list>. Print zones are not used. In either case, BASIC brackets any numeric values with one blank on each side. Negative numbers are prefaced by a minus sign rather than a blank. No blanks are added to string values.

LPRINT To advance one line on the Epson RX-100 printer (space up) without carriage return, enter BASIC statement:

LPRINT CHR\$(10);

or use "line feed" button (LF) on printer.

Entering just LPRINT gives a line feed—both a space up one line (line feed) and a return to left margin (carriage return).

LPRINT—Double Comma To leave space on the print line between items, put an extra comma (,,) in the print list. Enter BASIC statement:

LPRINT A,,B

This would print A in print zone 1, nothing in print zone 2 and put B in print zone 3,(col 28). See also Print Zones, Print Lines.

LPRINT—Normal Size Print To return to normal ten character-per-inch print size, you must turn off all non-standard print options now turned on. See Type Formats for complete instructions and examples.

LPRINT—Page Length To set page length to <n> lines-per-page, enter BASIC statement:

LPRINT CHR\$(27);"C";CHR\$(<n>)

<n> ranges between 1 and 127 and represents lines per page.

LPRINT—Print Zone Each group of 14 spaces across the print line is called a print zone. A comma (,) in an LPRINT list of items to be printed means “start printing the following item at the start of the next print zone.” Contrast this with the semicolon (;), which means the next item is to print immediately after this one, without even a single space between them. The print zones begin in columns 0, 14, 28, 42, 56, and 70. See also Print Lines.

LPRINT—Spacing in BASIC To get one or more spaces between fields printed by your BASIC programs, use a literal of spaces like: “ ”. To get several spaces between the printed values of A\$ and B\$ use:

```
LPRINT A$;“ ”;B$
```

See also Print Zones.

LPRINT—To Get Blank Line on Printer An LPRINT statement with no other specifications will print a blank line (that is, feed the paper up one line and return to left margin), so you can space your print-out format neatly.

LPRINT—6 Lines-Per-Inch To set the Epson RX-80 for this line spacing on the printer, enter BASIC statement:

```
LPRINT CHR$(27);“2”;
```

This is the default value for lines-per-inch in effect when you first power the printer on. See Type Formats for more information and examples.

LPRINT—8 Lines-Per-Inch To set the Epson RX-80 for this line spacing on the printer, enter BASIC statement:

```
LPRINT CHR$(27);“0”;
```

See Type Formats.

LPRINT USING BASIC Statement. LPRINT USING is used to format and print data on the printer. The format is:

```
LPRINT USING “<format string>”;  
                  <expression list>
```

<expression list> is a list of the numeric and/or string expressions that are to be printed. They may be numeric and string, variables and constants, and are separated by commas or semicolons.

The punctuation of the expression list determines the spacing of the items printed. The print line consists of print zones, fourteen spaces each. A comma in the expression list will cause the next item to be printed starting in the first space of the next print zone. A semicolon (;) will cause the next item to be printed immediately following the previous item.

In either case numeric values are bracketed by one

space on each end or followed by a blank and prefaced by a minus sign if the number is negative. No blanks are added to string values.

<format string> consists of one or more field specifiers which define the format to be used for printing <expression list> values as well as the type of data to expect in the <expression list>. The field specifiers are:

“!” Cues the program to expect string data and instructs it to print only the first character in the string. Example:

```
LPRINT USING “!”;“BOB”
```

Prints B.

“nnn” Where each n represents a space between the quote marks, cues the program to expect string data and print two characters from the string if there are no spaces between the quote marks. If there are spaces between the quotes then two string characters are printed plus as many more as there are blanks. Example:

```
LPRINT USING “ ”;“Robin”
```

Prints Robi (because there are two blanks between the quote marks and BASIC adds two).

“####” Cues the program to expect numeric data and print as many digits of it as there are hatchmarks (#). If there are fewer digits than marks, then the digits are padded with zeros to their left until they fill the field, effectively right-justifying the digits. If there are fewer digits than there are marks, then BASIC prints all the digits in the expression list preceded by a percent sign (%) to show that the number of digits has overflowed the field specifier. A decimal point may be located at any position in the format string. If the formatting specifies a number of digits to precede the decimal point, the digit is always represented with blanks if necessary. Numbers to the right of the decimal are rounded, if necessary, to print them in the format specified. Example:

```
LPRINT USING “#####”;34
```

Prints the number given in the expression list (34) to four spaces, padding the left with blanks. The result is two blanks then 34.

```
LPRINT USING “##”;3456
```

Prints %3456

“,” Placing a comma to the left of the decimal point in a format string causes a comma to be printed separating every third digit left of the decimal point. A comma also specifies another digit position. If a digit to the immediate left of a potential comma is blank, the comma is printed as a blank, too. When used, the comma must be sandwiched

LPRINT USING

in the format string between other field specifiers, such as \$, **, and #.

"t" Cues BASIC to expect numeric data and may be appended to other field specifiers at the beginning or end. It causes the value of the expression to be preceded or followed respectively by its algebraic sign (+ or -). If the value of the number printed is positive, then a plus sign is printed on the same side of the expression value as it is used in the field specifier. If the expression is negative then a minus sign (-) is used in the same way. Example:

LPRINT USING "+####";-45

Prints the number in the expression list (-45) to four digits and prefaces it with its algebraic sign, in this case a minus sign (-). The algebraic sign is not considered a digit. The result is two spaces then -45.

"-" Cues BASIC to expect numeric data and may be appended to other field specifiers at the beginning or end. If the algebraic value of the expression formatted is negative, then BASIC places a negative sign (-) on the same side of the number that the minus is on in the field specifier. If the expression formatted has a positive value, and the minus sign follows the rest of the format specifier, then BASIC prints a blank at the end, rather than a minus sign. If the value of the expression is positive, and the minus sign precedes the rest of the field specifier, then BASIC prints a minus sign first and then the number value of the expression as formatted by the rest of the field specifier. For example:

LPRINT USING "###-";-78

Prints (78-) one space and the number 78, directly followed by a minus sign.

LPRINT USING "###-";45

Prints (45) the number 45 right justified by a blank in three spaces, followed by a blank, caused by the minus sign in the field specifier. However:

LPRINT USING "-###";45

Prints (-45) the number 45, right justified by a blank in three spaces, and preceded by a minus sign from the field specifier.

*** Cues BASIC to expect numeric data and to right justify it in a given field by padding it with asterisks. It may be used in conjunction with other field specifiers such as "#". For example:

LPRINT USING "*****";26

Prints (**26). The number printed to three spaces and right justified using three asterisks. The first two asterisks printed on the left are produced by the two asterisks in the format string. The third is a space turned asterisk. The space which would have been produced by padding two digits into three spaces (##), is transformed instead into an asterisk

by the presence of the first two asterisks in the format string.

\$\$ Cues BASIC to expect numeric data and prints a dollar sign (\$) to the immediate left of the number within its otherwise specified format. Only one of the dollar signs prints. The second one prints as a blank to the left of the dollar sign. The double dollar (\$ \$) specifier may be used in conjunction with other field specifiers. For example:

LPRINT USING "\$ \$###";56

Prints (\$56) two blanks, one from an unprinted dollar sign, one as padding from the third hashmark (#). Then follow the printed dollar sign and the right justified two digit numeric value.

\$ Cues BASIC to expect numeric data and works like a combination of the "" and the "\$\$" field specifiers. The two leading asterisks in the field specifier are printed and also turn any leading blanks in the formatted number to asterisks. The dollar sign (\$) in the field specifier is printed to the immediate left of the numeric value. "***\$" will work in conjunction with other field specifiers. For example:

LPRINT USING "***\$###";45

Prints (***\$45) three asterisks followed by a dollar sign followed by the numeric value. Two of the three asterisks printed are from the two asterisks in the field specifier. The third is a space turned asterisk. The space which would have been produced, by padding two digits into three spaces (##), is transformed into an asterisk by the presence of the first two asterisks. The dollar sign is printed by the field specifier (\$) to the immediate left of the numeric value.

^ ^ ^ ^ Four carets are placed after the digit position characters to indicate exponential format. E+nn is printed in the four spaces allowed by the four carets. You may specify any decimal point position; significant digits are left-justified, adjusted on the exponent accordingly. One digit position left of the decimal point is used to print a space or a minus sign if a leading + or trailing +/- has not been designated.

LPRINT and LPRINT USING are similar to PRINT and PRINT USING, except the output goes to the printer.

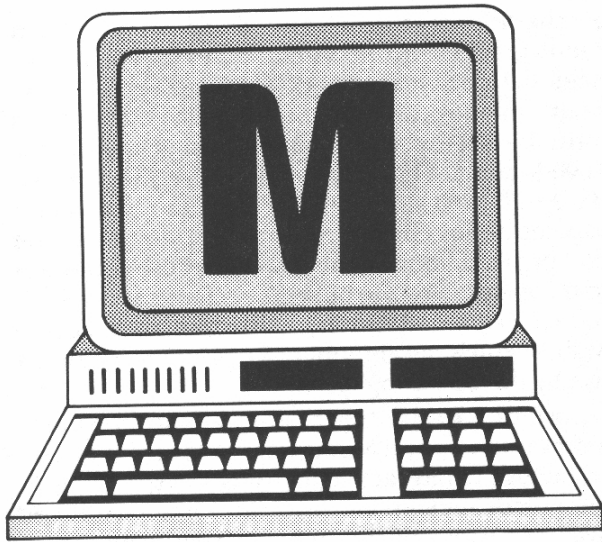
LPRINT assumes an 80-character-wide printer. BASIC automatically inserts a carriage return/line feed after printing 80 characters. Unless you end the statement with a semicolon, BASIC skips two lines when you print exactly 80 characters.

LPT: Reserved Device Name in BASIC—Stands for Printer. It is used in conjunction with the SAVE statement in BASIC to send data to the printer.

LRC Logitudinal Redundancy Check. See CRC.

LSB Least Significant Bit. An address storage format. The format is used for address storage by 80C85 and many other microprocessors. The four hexadecimal digits required to address a 64K memory are stored two per byte, but the order of the two pairs is reversed. Thus FEE0 is stored as E0 followed by FE. See Address Notation.

LXI 80C85 Assembly Language Instruction. Load register pair immediate. Byte 3 of the instruction is moved into the high-order register <rh> of the register pair <rp>. Byte 2 of the instruction is moved into the low-order register <rl> of the register pair <rp>. The addressing mode is immediate. No flags are set.



M Codes. ASCII 77, HEX 4D. **m**—ASCII 109, HEX 6D.

Machine Language Program All programs that run directly on the Model 100 are written in Machine language, the language of the actual numeric instruction code of the Model 100's 80C85 microcomputer chip. Most of these were originally written in higher level languages such as FORTRAN, COBOL, or BASIC. These were then translated by a compiler or an interpreter program into Machine language.

The object program contains the Machine language instructions for the Model 100's 80C85 which correspond to the instructions of the original source program. See Assembly Language.

Machine Language Program To load a Machine language RAM or cassette file to BASIC, the format is:

LOADM "[<dev>][<file.nm>]"

<dev> is the device that the file to be loaded is currently saved in. It may be a cassette (CAS:) or RAM (RAM:) file. If no device is specified then BASIC assumes the device is RAM.

<file.nm> is the name of the file that is to be loaded to BASIC. The extension (.nm), if used, should be .CO in the case of a RAM file, but it is usually not used at all.

Prior to loading a Machine language file, a segment of high memory is usually protected from use as BASIC workspace by using the CLEAR command. See CLEAR, Address Notation, CLOADM.

Machine Language Program—RUN To run a Machine language program use the BASIC com-

mands LOADM or CLOADM to load the program into BASIC high memory from the current location on cassette or in RAM. The CLEAR command protects this portion of BASIC high memory from being used by BASIC. When the Machine language program is loaded, the LCD displays the start and end addresses the program was saved to in high memory, as well as the entry address, if any. You may then run the program by using the BASIC CALL statement. See CLOADM, LOADM, and CALL.

Machine Language Program—Save with BASIC See SAVEM.

Machine Language Program Subroutine BASIC. See GOSUB and RETURN, RETURN ON MDM GOSUB, ON COM GOSUB, ON TIME\$ GOSUB, and ON...GOSUB.

Magazines A number of popular magazines contain useful information concerning the Model 100 and computers in general. Most computer stores and the larger bookstores carry a good assortment of these magazines. *BYTE* and *Infoworld* are excellent sources for general information on microcomputers. Currently the only magazine especially for the Model 100 is *Portable 100*.

If you want to subscribe or write for information, here are the addresses:

Breakthrough
Newsletter for Portable Computing
P.O. Box 230
Logan, Utah 84320
800-824-7888

Briefcase Portable
560 S. Hartz Ave. Suite 447
Danville, CA 94526

BYTE
Subscription Dept.
P.O. Box 590
Martinsville, NJ 08836
U.S. = 1 year \$19 12 issues

InfoWorld
Circulation Dept.
Box 837
Framingham, MA 01701
U.S. = 1 year \$25 52 issues

Portable 100
P.O. Box 468
Hasbraick Heights, NJ 07604

Mail Master* • Main Menu

PCM

Portable Computing Magazine
9529 U.S. Highway 42
P.O. Box 209
Prospect, KY 40059
502-228-4492

Portable Computer

Circulation Dept.
500 Howard Street
San Francisco, CA 94105
U.S. = 1 year 12 issues

80Micro

Subscription Dept.
POB 981
Farmingdale, NY 11737
U.S. = 1 year \$35.97 12 issues

Mail Master* Compatible with the MicroEditor program made by the same people, this program prints mailing labels, cross references, alphabetizes, and controls sorting. Includes a help feature too. Alphaware Incorporated.

Main Memory ROM and RAM together make up the internal, or main memory, of any computer. Memory is the ability to store and retrieve information. Memory, by itself, is usually a reference to RAM. This is the general purpose, erasable, and reusable memory located inside the Model 100.

ROM contains fixed data, usually programs such as the Model 100's BASIC ROMs, and its ROM operating system (BASIC Input/Output System). The ROM contains the fundamental Machine language programs to run the various devices attached to the Model 100, such as monitor, printer, diskettes, or cassettes. See Memory.

Main Menu A menu is a screen display which lists a number of possible options and asks the user to select one. A selection is made by entering an identifying number or letter, or positioning the cursor beside or on the desired item, using a light pen, cursor keys, etc. This may require either a branch or subroutine call to the code for the function, or the program to carry out this function may be loaded into memory and executed.

When you turn on the Model 100 you are in the Main Menu program which provides access to all five of the ROM resident application programs. The names of these built-in programs are listed on the Main Menu screen. They are: BASIC, an extended Microsoft BASIC interpreter; TEXT, a text processing program; Telcom, a telecommunications program; ADDRSS, an address organizer; and SCHEDL,

a schedule organizer. Also listed on the Main Menu are the current system values for the month, calendar day, year, day of the week, and time in hours, minutes, and seconds. Initially, these values must be set by you in BASIC using the DAY\$, TIME\$, and DATE\$ commands. Once they are set, the Model 100 automatically updates these values to keep them current, even if you don't use the computer, as long as the RAM memory continues to be powered by the built-in, rechargeable ni-cad (nickel-cadmium) battery. Also displayed on the Main Menu is the amount of RAM memory free for use by the operating system.

To use one of the five built-in applications programs listed on the Main Menu, move the cursor, an inverse video rectangle, using the space bar or the cursor movement keys, until it is positioned over the name of the program you want. Then, press ENTER. You are now in the chosen program. Any RAM files you create using TEXT or by saving from BASIC are also listed on the Main Menu and may be entered in the same way. TEXT files can be opened for viewing, modification, or printing. BASIC files that are entered from the Main Menu are automatically RUN. The two types of files are easily distinguished by name because TEXT uses the file extension .DO, indicating an ASCII document, and BASIC uses the file extension .BR, indicating a non-ASCII encoded BASIC file.

In addition to the names of the five programs permanently in ROM, there are nineteen spaces on the Main Menu available for listing the RAM files you create. Once all spaces on the Main Menu are taken by existing files, you cannot create any new ones. TEXT ignores your request for a new file name, beeps, and redisplay the "File to edit" prompt. Unless you enter BASIC to KILL an existing RAM file, thus making space for a new file, TEXT will only let you enter files that are already listed on the Main Menu. If all nineteen spaces for listing RAM files on the Main Menu are already taken and you try to create a new file by SAVEing a BASIC program, you are prompted with the message, "?FL Error", and undefined error.

To exit back to the Main Menu from any one of the five applications programs, press F8, usually defined as the Menu key. In some applications, such as Telcom terminal mode, it may be necessary to press F8 a second time before reaching the Main Menu.

Although normally the Main Menu welcomes you when you first turn on your computer, it is possible to define another program to run instead. To do this you would use the BASIC statement IPL which

autoruns a BASIC program of your choice as the initial program load. You could, for instance, write a program to protect the contents of your computer by requiring the user to enter a special password before using the rest of the system. You might even use the POWER statement in BASIC to automatically turn off the computer after a given number of wrong answers. See Initial Program Load.

Maintenance, Diskette Diskette maintenance (or, file management) refers to the keeping track of files on diskettes. This includes creating them, finding them by name, insuring that adequate free space is available, maintaining backups, and deleting files no longer needed. These functions are supported by the various DOS functions, but require thoughtful planning by the user to insure proper results.

Some DataBase Management Systems (DBMSs) attempt to automate part of the work of keeping track of files and diskettes. Some DBMSs maintain files of control and tracking data on other files and diskettes, providing alternatives to DOS functions.

Management See Business Manager Series, The.

Management of Files File management or diskette maintenance are terms describing keeping track of files on diskettes. See Maintenance, Diskette.

Manual A process done by hand, not automated or programmed or a reference book, booklet, or other document.

Manual Rack* A black plastic rack allowing easy access to six manuals. Nat Hellman, III, Inc.

Math Functions Table

Math Function	Operation Performed
ABS	Absolute value of a number is returned
ASC	ASCII code is found
ATN	Arctangent calculation is performed
CDBL	A number is converted to double-precision
CINT	Conversion to an integer
COS	Cosine calculation is performed
CRSLIN	Gives vertical line number, position of the cursor
CSNG	A number is converted to single-precision

Math Function	Operation Performed
EOF	End-of-file status is returned
ERL	Shows the line number where the last error occurred
ERR	The error code of the last error is shown
EXP	Exponential calculation
FIX	Convert to integer by truncation
FRE	Gives the number of unused bytes of RAM memory
HIMEM	Returns highest memory address available to BASIC
INP	Reads a value from the CPU port
INSTR	Searching for a substring a string
INT	Integer conversion
LEN	Find string length
LOG	Natural logarithm is calculated
LPOS	Gives carriage position of the printer
MAXRAM	Gives the highest RAM memory address, size
PEEK	Memory address value is shown
POS	Gives the current cursor column position
RND	Pseudo-random number is given
SGN	Gives sign, in algebraic form
SIN	Sine is calculated
SQR	Square root is calculated
TAB	Cursor or print head positioning
TAN	Tangent is calculated
VAL	Returns the numeric value of the string
VARPTR	Memory address of a variable is given

Math Operator Table

Math Operator	Operation Performed
+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Integer Division
^	Exponentiation
MOD	Modulus Arithmetic

Mathematical Hierarchy of Operations Table • Memory

Mathematical Hierarchy of Operations Table

Highest Parenthesis (inner to outer)

^
+ , - (unary plus and minus)
* , /
MOD
+ , -
< > , = , > , < = , > , <

NOT

OR

XOR

EQV

IMP

Lowest

MAXFILES BASIC. A system variable that determines the number of files you can use in BASIC. Valid values that can be assigned to MAXFILES range between one (1) and fifteen (15). If you have not set the value of MAXFILES, then BASIC supplies the default value of one (1). If you try to read a value into MAXFILES that is out of its acceptable input range, (1-15), then you get a "?FC ERROR," an illegal function call message. For example:

MAXFILES=<num>

makes the maximum number of files in BASIC equal to the value of <num>. <Num> must be a value between one and fifteen. The value of max-files may also be displayed using a PRINT statement. See OPEN.

MAXRAM BASIC. A system constant that contains the highest memory address available to BASIC. The value of MAXRAM is the same no matter what the RAM memory size of your Model 100 is. The constant value of MAXRAM is 62960. See also HIMEM and CLEAR.

MC Machine Code. Actual Machine language instructions, whether written directly in MC or resulting from the translation of a source program.

MDM ON/OFF/STOP Enables, disables, or holds the modem communications interrupt previously defined by an ON MDM statement. The formats are:

MDM ON

or

MDM OFF

or

MDM STOP

MDM ON must be executed to activate the ON MDM statement. BASIC checks before each statement is executed to see if the built-in modem has received any characters. If so, a GOSUB is executed

to the line specified in the ON MDM statement. See ON MDM GOSUB.

In MDM OFF, no trapping takes place for the modem; however, characters being received are remembered, and a trap takes place immediately upon the next execution of a MDM ON statement.

Media Materials or devices used for recording and storing information are called media. The main medium for the Model 100 is RAM, but cassette, ROM diskette, and other devices are also used. Media are often classified as removable media, such as diskettes and cassettes, or as fixed media, such as RAM and ROM. Fixed media are not removable from the device that drives them, so there is no ability to store additional data or backup copies off-line (outside the computing system) for insertion when needed.

Mediamix ETI(2)* The Mediamix ETI(2) converts an IBM typewriter into a letter-quality printer, enabling the typewriter to function as a computer printer or a manual typewriter. Applied Creative Technology, Inc.

Memory The capacity to store and retrieve information. The Model 100 relies primarily on random access memory (RAM), read-only memory (ROM), diskettes, and cassettes. Memory, by itself, is usually a reference to RAM. This is the general purpose, erasable, and reusable memory located inside the Model 100.

ROM (read-only) memory contains fixed data, usually programs such as the Model 100's BASIC ROMs and its ROM operating system (BASIC Input/Output System). The ROM contains the fundamental Machine language programs to run the various devices attached to the Model 100, such as monitor, printer, diskettes, cassettes, etc. ROM and RAM together make up the internal memory or main memory of the Model 100, or any other computer.

Contrast this with external memory—such as cassette, diskette, and hard disk—which involves mechanical motion to retrieve data and is thus hundreds or thousands of times slower than internal memory. Data in internal memory is immediately available to programs for processing. Data in external memory must be copied into internal memory (READ or INPUT), processed, then copied back out to external memory (WRITE or OUTPUT).

If data has been created, it can be written out to external memory without a READ first. And, if data read in from external memory has not been modified, there is no need to write it back out since the original copy is still there.

While external memory is very slow relative to internal memory, it also has advantages. It is much cheaper per character of data stored on-line (available for processing without manual intervention). In addition, the ability to store external memory data off-line (such as diskettes or cassettes in a box) allows essentially unlimited storage of data. This of course requires a manual step of inserting the diskette/cassette/disk before the data can be loaded into the internal memory for processing. See also Virtual Memory, RAM Disk, Spooling, Bank Switching, Memory Maps, Memory Addresses, Memory Segments, Bubble Memory, RAM Memory Cards, Diskette Formats.

Memory Change with BASIC. See POKE.

Memory Read byte from BASIC. See PEEK.

Memory The total free RAM memory amount in bytes is shown on the Model 100 Main Menu. The format is:

“nnnn bytes free”

where nnnn is the current number of bytes free in RAM. This does not equal the amount available for use by you, however, because the operating system may need to use some of the free RAM too. Usually RAM memory is available to the user with “195 bytes free” on the menu.

Memory, Amount Free BASIC. See FRE.

Memory, Protect From BASIC See CLEAR.

Memory Storage in Bytes Byte is a label for storage to hold one character (letter, digit, etc.) in computer memory, internal or external. Abbreviated B. In thousands, KB or simply K. (Actually, 1K = 1024 bytes, because this number is an even power of 2). The Model 100 is available in 8, 16, 24, and 32K RAM memory sizes. See Address.

Memory Address A number or variable designating a location in memory. See Address.

Memory Card A card containing RAM or ROM memory to expand or enhance the computer's main memory.

Memory Dump Program The following program prints out the contents of any range of memory addresses. It gives the memory address, the ASCII code residing there, the 7-bit (readable) character representation of that code, and the 8-bit (graphics) character representation of that code if applicable. If the ASCII value in a memory address is not in the 8-bit range then the 8-bit column repeats the character in the 7-bit column. Since the 7-bit ASCII codes below 33 are control codes and

unprintable, this program uses the notation CTRL to indicate these values.

The output from this program may be displayed on the LCD screen or on an attached printer. If you choose the onscreen display then you have the option of continuous scrolling, which you can pause at any time using the BREAK/PAUSE key. Or, you may choose to have the display stop at the end of each screenful, pausing until you resume output by pressing any key on the keyboard. If you choose a printer display, not only will the 7-bit codes between zero and 33 display as CTRL, but also the 8-bit values between 128 and 161. This is because the printer may respond to these 8-bit codes as if they were 7-bit codes (i.e., control codes) with various strange results. Since different printers have different graphics capabilities, printer responses to the other 8-bit ASCII codes will vary, displaying sometimes the 7-bit equivalent character, sometimes an 8-bit graphics character, and sometimes no character at all.

```

10 CLS
20 PRINT "MEMORY DUMP PROGRAM
   Copyright Gary Phillips and Associates"
30 PRINT
40 INPUT "START ADDRESS";ST: INPUT "END
   ADDRESS";ND
50 PRINT
60 PRINT "THIS PROGRAM WILL DISPLAY
   OUTPUT ON THE LCD SCREEN OR ON AN
   ATTACHED PRINTER."
70 INPUT "PRINTER (P) OR SCREEN (S)";M$:
   CLS
80 IF M$="P" OR M$="p" THEN M=1 ELSE
   M=2
90 IF M=2 THEN PRINT "THE ONSCREEN DIS-
   PLAY ALLOWS TWO OPTIONS: 1) SCROLL
   CONTINUOUSLY AND STOP AND START
   THE DISPLAY AT WILL USING THE PAUSE
   KEY. 2) PAUSE THE DISPLAY AUTOMATI-
   CALLY WITH EACH SCREENFUL AND PRESS
   ANY KEY TO GO TO THE NEXT SCREENFUL."
100 PRINT
110 IF M=2 THEN INPUT "SCROLL (S) OR
   PAUSE (P)";P$: CLS
120 IF P$="P" OR P$="p" THEN P=2 ELSE P=1
130 IF M=1 THEN LPRINT: LPRINT "ADDRESS
   "; "7BIT "; "ASCII "; "8BIT"
140 FOR I=ST TO ND
150 IF M=2 AND P=2 OR M=2 AND 1=ST THEN
   PRINT "ADD "; "7BT "; "8BT "; "ASCII"
160 IF M=2 AND P=2 THEN FOR SC=1 TO 6
170 IF I≠0 AND SC=1 THEN I=I-1
180 LTR=PEEK(I)

```

Memory Map for the Model 100

```

190 LR=LTR
200 IF LTR>128 THEN LTR=LTR-128
210 IF LTR<33 THEN C=1 ELSE C=2
220 IF LR<33 OR LR=127 OR LR=224 THEN C=3
230 IF M=1 AND C=2 THEN LPRINT I;" ";
CHR$(LTR); " "; LR; " "; CHR$(LR)
240 IF M=1 AND C=1 THEN LPRINT I;"CTRL";
LR; " CTRL"
250 IF M=2 AND C=2 THEN PRINT I; " ";
CHR$(LTR); " "; CHR$(LR); " "; LR
260 IF M=2 AND C=3 THEN PRINT I; "CTR";
" CTR "; LR
270 IF M=2 AND P=2 THEN I=I+1
280 IF M=2 AND P=2 THEN NEXT SC
290 IF M=2 AND P=2 THEN A$=INPUT$(1)
300 NEXT I

```

Memory Map for the Model 100 In the Model 100, ROM occupies an address range in low memory. ROM addresses range between 0000 and 7FFF (hexadecimal). RAM memory may be located between memory addresses E000 (hexadecimal) and DFFF (hexadecimal). If you have a full 32K of RAM then all addresses will be occupied. If you have less than the maximum, then this segment of memory is occupied starting at the highest address and fills in downwards to occupy the lower address ranges of this RAM segment. If you have the minimum 8K of RAM, then your RAM memory resides between address locations E000 (hexadecimal) and FFFF (hexadecimal).

No matter what size RAM your machine has, it uses the same communications area, between locations F5F0 hexadecimal (or 62960 decimal) and FFFF hexadecimal (or 65535 decimal). This range in high RAM is reserved for system use. You may loose data if you POKE values into this segment of high memory. RST means reset Machine language.

Memory Location		
Decimal	Hexadecimal	Functions
0	0000	Begin BASIC ROM RST 0. Jumps for initialization (IPL) at address 7F33.
8	0008	RST 1. Syntax check. Normally executed immediately following a RST 2. Checks the syntax of the byte directly after the RST 1 instruction to see if it matches the byte pointed to by the HL register. If there is no match, execution branches to an error handling subroutine. If they do match, execution jumps to the location two bytes after the RST 1 instruction.

Memory Location		
Decimal	Hexadecimal	Functions
16	0010	RST 2. Parses the current BASIC line. Puts the first meaningful character from the BASIC buffer into the A register and its address into the HL register pair.
24	0018	RST 3. Compares DE and HL and sets the flags.
32	0020	RST 4. Displays the contents of the A register on the LCD.
36	0024	Trap interrupt routes execution to a power down sequence (via F602 to 1431).
40	0028	RST 5. Execution jumps to 1069.
44	002C	RST 5.5. A Hook for coming expansions. Disables all interrupts and branches execution to 62696 (decimal) or F5F9 (hexadecimal) where the interrupts are reenabled. Execution then returns to this address.
48	0030	RST 6. Execution jumps to 13276 (decimal) or 33DC (hexadecimal).
52	0034	RST 6.5. Disables all interrupts and branches execution to 28076 (decimal) or 6DAC (hexadecimal). Currently that address contains a RET which returns execution to this address. Looks like another hook.
56	0038	RST 7. Branches execution to 32726 (decimal) or 7FD6 (hexadecimal). It then uses the byte immediately following the RST instruction to calculate an offset into a table beginning at 64218 (decimal) or FADA (hexadecimal). The resulting table value determines further branching, with execution returning in the end to the second byte following the RST. Currently the branching caused by the table does not perform any useful code; rather it seems to be another hook for future use.
60	003C	RST 7.5. Disables all interrupts and branches execution to 6962 (decimal) or 1B32 (hexadecimal). Once again there is an unused hook at address location F5FF which sets the interrupt mask at 0D and reenables the interrupts.

Memory Map for the Model 100

Memory Location		
Decimal	Hexadecimal	Functions
128	0080	Beginning of the keyword list. In each case the first character of the first word has a set value for its highest order bit. The address at which a keyword is located is equivalent to its BASIC token value.
BASIC Keyword List		
129	0083	FOR
130	0086	NEXT
131	008A	DATA
132	008E	INPUT
133	0093	DIM
134	0096	READ
135	009A	LET
136	009D	GOTO
137	00A1	RUN
138	00A4	IF
139	00A6	RESTORE
140	00AD	GOSUB
141	00B2	RETURN
142	00B8	REM
143	00BB	STOP
144	00BF	WIDTH
145	00C4	ELSE
146	00C8	LINE
147	00CC	EDIT
148	00D0	ERROR
149	00D5	RESUME
150	00DB	OUT
151	00DE	ON
152	00E0	DSKO\$
153	00E5	OPEN
154	00E9	CLOSE
155	00EE	LOAD
156	00F2	MERGE
157	00F7	FILES
158	00FC	SAVE
159	0100	LFILES
160	0106	LPRINT
161	010C	DEF

Memory Location		
Decimal	Hexadecimal	Functions
162	010F	POKE
163	0113	PRINT
164	0118	CONT
165	011C	LIST
166	0120	LLIST
167	0125	CLEAR
168	012A	CLOAD
169	012F	CSAVE
170	0134	TIME\$
171	0139	DATE\$
172	013E	DAY\$
173	0142	COM
174	0145	MDM
175	0148	KEY
176	014B	CLS
177	014E	BEEP
178	0152	SOUND
179	0157	LCOPY
180	015C	PSET
181	0160	PRESET
182	0166	MOTOR
183	016B	MAX
184	016E	POWER
185	0173	CALL
186	0177	MENU
187	017B	IPL
188	017E	NAME
189	0182	KILL
190	0186	SCREEN
191	018C	NEW
192	018F	TAB
193	0193	TO
194	0195	USING
195	019A	VARPTR
196	01A0	ERL
197	01A3	ERR
198	01A6	STRING\$
199	01AD	INSTR
200	01B2	DSKI\$

Memory Map for the Model 100

Memory Location		
Decimal	Hexadecimal	Functions
201	01B7	INKEY\$
202	01BD	CSRLIN
203	01C3	OFF
204	01C6	HIMEM
205	01CB	THEN
206	01CF	NOT
207	01D2	STEP
208	01D6	+
209	01D7	-
210	01D8	*
211	01D9	/
212	01DA	^
213	01DB	AND
214	01DE	OR
215	01E0	XOR
216	01E3	EQV
217	01E6	IMP
218	01E9	MOD
219	01E6	\
220	01E9	>
221	01EC	=
222	01ED	<
223	01EE	SGN
224	01EF	INT
225	01F0	ABS
226	01F3	FRE
227	01F6	INP
228	01F9	LPOS
229	01F9	POS
230	01FF	SQR
231	0203	RND
232	0206	LOG
233	0209	EXP
234	020C	COS
235	020F	SIN
236	0212	TAN
237	0215	ATN
238	0218	PEEK
239	021B	EOF

Memory Location		
Decimal	Hexadecimal	Functions
240	021E	LOC
241	0222	LOF
242	0225	CINT
243	0228	CSNG
244	022B	CDBL
245	022F	FIX
246	0233	LEN
247	0237	STR\$
248	023A	VAL
249	023D	ASC
250	0241	CHR\$
251	0244	SPACE\$
252	0247	LEFT\$
253	024B	RIGHT\$
254	0251	MID\$
255	0256	'

End BASIC Keyword List

610	0262	Beginning of the Verb Table. The Verb Table contains the ROM addresses (in LSB MSB format) of the Machine language routines used to execute each BASIC keyword. To find this address for a keyword, take the BASIC token value (the address at which the keyword is contained within the keyword list) and subtract 128 (decimal) or 80H (hexadecimal) from the token value. Then multiply the result by two. Use the resulting number as an offset into the verb table which stores the memory addresses to which execution must jump, in order to execute the keywords.
796	031C	Begin BASIC error codes
859	035B	End BASIC error codes
916	0394	Begin cold start values for RAM
1024	0400	End cold start values for RAM
1890	0762	FOR
2162	0872	DEF
2319	090F	RUN
2334	091E	GOSUB
2358	0936	GOTO
2406	0966	RETURN

Memory Map for the Model 100

Memory Location		
Decimal	Hexadecimal	Functions
2462	099E	DATA
2464	09A0	REM
2499	09C3	LET
2607	0A2F	ON
2736	0AB0	RESUME
2831	0B0F	ERROR
2842	0B1A	IF
2894	0B4E	LPRINT
2906	0B56	PRINT
3141	0C45	LINE
3235	0CA3	INPUT
3289	0CD9	READ
3878	0F26	Routine that determines execute addresses for string functions. It compares the keyword parsed with a list of possible values and then jumps to the address associated with that keyword when a match is found.
4364	110C	OUT
4411	113B	LLIST
4416	1140	LIST
4747	128B	POKE
4811	12CB	Routine which shows a cursor on the LCD screen and then waits for keyboard input. The ASCII value of the first character input is assigned to the A register. This routine is called by the Machine language routine for INKEY\$, located at 19386 (decimal) or 4BEA (hexadecimal).
5083	13DB	Routine waits for keyboard input and assigns the first character value input to the type-ahead buffer.
5145	1419	POWER
6539	198B	TIMES
6589	19BD	DATE\$
6641	19F1	DAY\$
6776	1A78	IPL
6814	1A9E	COM and MDM
6962	1B32	RST 7.5 causes execution to jump to this address.
7069	1BB8	KEY

Memory Location		
Decimal	Hexadecimal	Functions
7255	1C57	PSET
7366	1C66	PRESET
7619	1DC3	ELSE
7621	1DC5	SOUND
7660	1DEC	MOTOR
7674	1DFA	CALL
7714	1E22	SCREEN
7774	1E5E	LCOPY
7778	1E62	Transfers the contents of the LCD display buffer to the LCD screen.
7961	1F19	KILL
7994	1F3A	FILES
8247	2037	NAME
8446	20FE	NEW
8832	2280	CSAVE
9079	2377	CLOAD
13276	33DC	RST 6 branches execution to this address
16384	4000	Applications ROM
16511	407F	RESTORE
16538	409A	STOP
16543	409F	END
16602	40DA	CONT
16633	40F9	CLEAR
16756	4174	NEXT
16937	4229	BEEP
16945	4231	CLS
18315	478B	DIM
19386	4BEA	INKEY\$
19659	4CCB	OPEN
19824	4D70	LOAD
19825	4D71	MERGE
19919	4DCF	SAVE
20008	4E28	CLOSE
20591	506F	LFILES
20593	5071	DSKO\$
22423	5797	MENU
24145	5E51	EDIT

Memory Map for the Model 100

Memory Location		
Decimal	Hexadecimal	Functions
28076	6DAC	RST 6.5 disables interrupts and branches execution to this address where it again branches to 62972 (decimal) or F5FC (hexadecimal). That address contains a RET as a hook for future use, which returns execution to this address.
32051	7D33	RST 0 branches execution to this routine which initializes system values and disables interrupts.
32052	7D34	Sets the stack pointer to 64448 (decimal) and FCC0 (hexadecimal).
32055	7D37	Begin delay routine.
32063	7D3F	End delay routine.
32064	7D40	Begin 8155 I/O port initialization.
32083	7D53	End 8155 I/O port initialization.
32523	7F0B	MAX
32726	7FD6	RST 7 causes execution to branch to this address
32767	7FFF	ROM upper address limit
Begin RAM		
32768	8000	Begin Fourth 8K RAM (24K-32K)
40959	9FFF	End Fourth 8K RAM (24K-32K)
40960	A000	Begin Third 8K RAM (16-24K)
49151	BFFF	End Third 8K RAM (16K-24K)
49152	C000	Begin Second 8K RAM (8K-16K)
57343	DFFF	End second 8K RAM (8K-24K)
57344	E000	Begin First 8K of RAM (only 5K for user)
62960	F5F0	End First 8K of RAM; also MAXRAM system constant, the highest memory address available to BASIC.
62961	5F51	Begin Overhead for the operating system
62964	F5F4	Points to HIMEM
62969	F5F9	RST 5.5 branches execution to this address after disabling all interrupts. This address re-enables the interrupts and returns to 44 (decimal) and 002C (hexadecimal).

Memory Location		
Decimal	Hexadecimal	Function
62972	F5FC	RST 6.5 branches execution to this address which currently contains a RET, returning execution to 52 (decimal) or 0034 (hexadecimal)
62975	F5FF	RST 7.5 branches execution to this address after disabling all interrupts. This address contains a RET which returns execution to 60 (decimal) or 003C (hexadecimal).
62978	F602	Trap interrupt vector
63067	F65B	Parameter image for the RS-232C interface
63109	F685	Contains the most recent input to the computer as well as the program line that comes after the input. Begin keyboard input buffer.
63113	F689	Contains the default function key definitions for BASIC.
63358	F77E	End Keyboard input buffer
63369	F789	Begin BASIC function keys
Files		
63385	F799	LOAD"
63401	F7A9	SAVE"
63417	F7B9	RUN
63433	F7C9	LIST
63449	F7D9	()
63465	F7E9	()
63481	F7F9	MENU
63496	F808	End BASIC function keys
63498	F80A	Begin repeat of 63369-63496 address locations.
63625	F889	End Repeat of 63369-63496 address locations.
63841	F961	Begin menu display. Each file is represented by eleven bytes in this segment of memory. The first byte shows file type (00 = empty, C0 = ASCII file, 80 = tokenized BASIC file, A0 = Machine language file and B0 = ROM file). The second byte is the LSB of the file start address. The third byte is the MSB of the file start address. The remainder of the eleven bytes are the file name and extension in ASCII format.

Memory Power Switch • MERGE

Memory Location		
Decimal	Hexadecimal	Function
BASIC		
63853	F96D	TEXT
63864	F978	TELCOM
63875	F983	ADDRSS
63886	F98E	SCHEDL
63897	F999	Suzuki Hiyashi
63908	F9A4	()
64128	FA80	End menu display
64201	FAC9	Begin buffer space used by RST 7 in calculating an offset into the branch table at 64218 (decimal) or FADA (hexadecimal).
64202	FACA	End buffer space used by RST 7.
64218	FADA	Begin branch table used by RST 7. At present all entries in the table point to C9 which contains a RET. This returns execution to 58 (decimal) or 003A (hexadecimal).
64363	FB6B	String stack for BASIC.
64404	FB94	Pointer indicates current DATA line used by BASIC.
64414	FB9E	Top of the BASIC stack
64416	FBA0	ERL
64428	FBAC	CONT
64436	FBB4	Contains the address of the next memory location available to BASIC.
64438	FBB6	The next BASIC variable.
64440	FBB8	DATA line between-item pointer
64442	FBBA	Table of Variable definitions
64448	FCC0	Stack
64904	FD88	Begin menu time display.
64927	FD9F	End menu time display.
65024	FEE0	Begin LCD screen image copy in RAM.
65343	FF3F	End RAM copy of LCD screen image.
65344	FF40	Begin video buffer for TELCOM.
65535	FFFF	End video buffer for TELCOM.
End Overhead for the Operating System		

Memory Power Switch The memory power switch is located on the underside of the Model 100. If the memory power switch is Off, the computer won't operate. This is true even if the On/Off switch is On and the Model 100 is connected to an external power supply, such as AA batteries or wall current (via the AC/DC adapter unit.) Turning the memory power switch Off erases the contents of RAM memory by disconnecting it from its power supply, the built-in rechargeable nicad battery.

You must turn on all power switches and supply power from an external source (such as AA batteries or AC/DC adapter) before the computer will run. The built-in nicads are automatically recharged by the same DC current that is operating the computer. When you turn the On/Off switch Off, this turns the computer off but does not effect the contents of RAM memory, which continues to be powered by the nicad battery as long as the memory power switch is On.

Memory Segment A continuous block of memory addresses, such as 0 to 16K.

Menu A menu is a screen display which lists a number of possible options and asks the user to select one. A selection is made by keying in an identifying number or letter or positioning the cursor beside the desired item, using a light pen, etc. This may require either a branch or subroutine call to the code for the function; alternately the program to carry out this function may be loaded into memory and executed. See Main Menu.

MENU BASIC. Exits from BASIC and returns the user to the Main Menu. The command format is:

MENU

The MENU command in BASIC mode has the same effect that pressing F8 has in all the application programs in the Model 100.

Menu-Driven See Menu.

MERGE BASIC Command. MERGE is used to add lines from an ASCII program file into the program currently in BASIC memory. The formats are:

MERGE ["<dev>:"] ["<filename>"]

or

MERGE "<dev>:[<r>]<w> <p> <s>"

<dev> specifies the device from which the ASCII merge file is to be merged with the current program in BASIC memory. Merge files may be read and merged to BASIC from the following devices. RAM: an ASCII format file stored in RAM memory. CAS: an ASCII format file stored on cassette tape.

MERGE • Message

COM: a communications file received via the RS-232C interface.

MDM: a communications file received via the built in modem.

<filename> is a string expression naming the file stored in ASCII format. It is used with the CAS: and RAM: devices.

If the device <dev> is RAM or CAS, the device is searched for the named file. If the device is MDM: or COM:, the incoming communications file is used. In this case, the sender must use the same configuration of communications parameters you specify. The program lines in the device file are merged with the program lines in BASIC memory. If any lines in the ASCII file being merged have the same line number as a line in the program in BASIC memory, the line from the file replaces that line in BASIC memory. The MERGE command puts the merged program in BASIC memory and returns BASIC to command level.

If the device name is omitted, RAM: will be assumed as the device. When CAS: is specified as the device, you may omit the file name, and BASIC will MERGE the first ASCII file encountered on the cassette tape. When RAM: is specified as the device, you must specify <filename>, although the two-character extension used in RAM is optional. If you do omit the extension, and if there are two files with the same name but with .BA and .DO extensions respectively, BASIC will merge the .BA file, thereby creating an error condition. Only .DO files may be merged.

A device specification of MDM: or COM: indicates that the incoming file is a communications file. To merge a communications file, use the second command format, which specifies a list of communications parameters [<R>]<w><p><s> rather than a <filename>. The communications parameters you specify in the MERGE command should be the same ones the sending party is using to transmit the communications file to you via the modem or RS-232C port. Each communications parameter is represented by a single-digit value. <r> is the baud rate. Valid values range between 1 and 9:

- 1 = 75 baud
- 2 = 110 baud
- 3 = 300 baud
- 4 = 600 baud
- 5 = 1200 baud
- 6 = 2400 baud
- 7 = 4800 baud

8 = 9600 baud

9 = 19200 baud

If the communications file to be merged is specified as coming from the modem (MDM:), then the <r> communications parameter value should be omitted. The modem default rate is 300 baud.

<w> represents word length in bits. May be set to values 6, 7, or 8.

<p> represents parity. May be set to E (meaning even), O (meaning odd), I (meaning ignore parity), N (meaning no parity).

 indicates stop bits—1 for one stop bit, or 2 for two stop bits.

<s> enables and disables XON/OFF status. Valid values are E to enable it, and D to disable it.

MERGE With a SAVED BASIC Program named "SAMPLE":. If you want to save the program in ASCII format, which allows you to MERGE two programs, put ,A after the close quote of the program name; thus:

SAVE "[<dev>:]<filename>,A]"

<dev> may be RAM: or CAS:. If omitted, BASIC assumes the device is RAM:.

<filename> is the name you give the program saved in ASCII format from BASIC. This is the file name you use when you specify a merge file.

The SAVE instruction does not alter your program in BASIC memory. If you write a BASIC program, it is erased unless you SAVE it before you turn off the Model 100, LOAD a program, or use NEW. See also BASIC EDITing.

The instructions for SAVEing to RAM or on cassette in ASCII format are as follows:

SAVE "SAMPLE",A

On cassette, first rewind the cassette, press PLAY and RECORD together, then enter:

SAVE "CAS:SAMPLE",A

To merge the program at a later time with the BASIC program currently in BASIC memory, use the MERGE command to copy it from the cassette or RAM file you saved it to back into the Model 100 memory. The program lines in the ASCII file given in the MERGE statement are merged numerically with the program lines in BASIC memory. In the case of two lines with the same number, BASIC chooses to keep the line from the ASCII merge file rather than the program line in BASIC memory.

Message A statement or code printed out or displayed on the screen by a program to let you know what is happening. Examples include prompts, error messages, and informational messages.

Messages, Error BASIC.

Error Message	Error Number	Error Condition
?AO Error	53	File Already Open
?BN Error	51	Bad File Number
?BS Error	9	Bad Subscript
?CF Error	58	Closed File Accessed
?CN Error	17	Can't Continue
?DD Error	10	Doubly Dimensioned Array
?DS Error	56	Direct Statement in File
?EF Error	54	Input Past End of File
?FC Error	5	Function Call Illegal
?FF Error	52	File Not Found
?FL Error	57	Undefined
?ID Error	12	Illegal Direct
?IE Error	50	Undefined
?IO Error	18	Error
?LS Error	15	String Too Long
?MO Error	22	Missing Operand
?NF Error	1	NEXT without FOR
?NR Error	19	No Resume
?NM Error	55	File Name Bad
?OD Error	4	Out of Data
?OM Error	7	Out of Memory
?OS Error	14	Out of String Space
?OV Error	6	Overflow
?RG Error	3	RETURN without GOSUB
?RW Error	20	Resume without error
?SN Error	2	Syntax Error
?ST Error	16	String Too Complex
?TM Error	13	Type Mismatch
?UE Error	21, 23-49, 59-255	Undefined Error
?UL Error	8	Undefined Line

MICROCOMPUTING* MICROCOMPUTING is a monthly general interest computer magazine. It is particularly geared for the novice computer enthu-

siast. Microcomputing Magazine/Wayne Green Inc.

MicroEditor II* Format your text files with the features offered by this text processing program. It creates right and left margins as well as justifying or centering your text. You control the vertical and horizontal dimensions of the text on the page and may optionally number both pages and lines. This program handles headings and form letters, and has tab and memory clear functions. A full-function help feature makes this package easy to use. Alphaware Incorporated.

Microprism* Microprism is a low cost printer with 75 CPS operation, pin and friction feed, and data printing at 110 CPS. It contains both RS-232 and parallel interfaces, and features data plot graphics. Integral Data Systems, Inc.

MID\$ When used as a function MID\$ returns the indicated part of a specified string. The format is:

`<strvr$> = MID$(<strex$>,<stpos>[,<lenth>])`

`<strex$>` must be a string expression.

`<stpos>` an integer expression range (1 to 255), indicating the location within the string of the first character to be returned.

`<lenth>` an integer expression range (0 to 255), indicating the number of characters, i.e., the length of the string to be returned.

The MID\$ function returns a string with a specified character length, `<lenth>`, which starts at `<stpos>`, which represents the numeric offset from the first string position to the first character to be returned from the string `<strex$>`. If the length of the string, `<lenth>`, exceeds the number of characters to the right of the start position, `<stpos>`, or the length value is omitted, all rightmost characters beginning with the start position, `<stpos>`, specified is greater than the length of the string, `<strex$>`, a null string is returned by the MID\$ function.

When used as a statement, MID\$ replaces a portion of one string with another string. The format is:

`MID$(<strx1$>,<stpos>[,<lenth>]) = <strx2$>`

`<strx1$>` is a string variable that will have some of its characters replaced.

`<stpos>` is an integer expression; range 1-255, indicating the location of the first character, within the first string `<strx1$>` to be replaced by the first character of the second string `<strx2$>`.

`<lenth1$>` is an integer value range 0 to 255, indicating the length of the string in `<strx1$>` to be replaced. It is optional and always has the same value as the length of `<strx2$>` regardless of what you may specify.

Mikrokolor Color Graphics Interface* • Modem Cable, Installing

<strx2\$> is any string expression which will replace some or all of the characters in <strx1\$>.

The characters in the first string expression, starting at the position specified by <stpos>, are replaced by the characters in the second string, <strx2\$>. As long as it fits, the entire second string replaces the portion of the first string that lies between the start position and the end. Any part of the second string that is too long to fit in the designated portion of the first string, is truncated to fit.

Mikrokolor Color Graphics Interface* Interface the Model 100 with a standard color television modulator, or color monitor using this interface that comes fully assembled and tested or in kit form. It provides high resolution (256 by 192) color graphics and text displays in fifteen colors, including transparent. This interface operates in four modes: Text, 6 by 8 dot-matrix characters, 40 characters by 24 lines, and 256 user-definable characters; Graphics, 8 by 8 matrix, 24 line by 32 character display, two-color characters, and 256 by 192 color graphics; Graphics2, easy-to-program graphics and animation for graphs, charts, games, and other displays, with sprites prioritized in thirty-two 3-D slide planes for display in fifteen colors and transparent; and Multicolor, 64 by 48 color graphics. The unit contains a built-in display processor with 16K of RAM and doesn't use any of the Model 100 memory. Software is included for text translation to the 24-line monitor with cursor control and automatic scrolling. The interface may also be used with a monochrome monitor to produce sixteen gray levels. Full documentation included. Andreasen's Electronics Research & Development, Inc.

Mind Master* Test wits with the Master in this real time game featuring sound and graphics effects. Alpine Data Systems.

Mixed Program Lines Old and NEW Programs. Erasing BASIC Memory. To start a new program type:

NEW

This completely erases all lines now in BASIC's memory, so if there are program lines in memory you want to keep and you haven't already stored them on cassette or in RAM, SAVE them first. If you don't erase the program in memory before starting on another, you will usually wind up with the unusable combination of mixed lines from your old and new programs.

Mnemonic A name or abbreviation which is intended to remind you of what it stands for, such as LAX for Load Accumulator into X register.

Model 100 Instrument Trainer* Pilots train on this realistic flight simulator that analyzes your performance with a postflight playback of glidepath and ground track. Includes a realistic simulation of the following practice approaches: ILS, ADF, Back Course, Localizer, VOR/DME, and Tacan. Instructions included. 24K. Pocket Programs.

Model 100 Special Interest Group See Compuserve.

Modem MOdulator DEModulator. A device to convert the digital voltage level signals of the computer to modulated (frequency or tone) signals for transmission over phone lines. Required at both ends of a phone line for telecommunications. The Model 100 has a built-in modem that operates at 300 baud. Other communications parameters may be set from within BASIC using the SAVE, LOAD, and MERGE commands, or directly in Telcom Entry mode. The built-in modem is switch selectable for answer or originate transmission frequency ranges. Setting the ANS/ORIG switch on the left side panel of the Model 100 to ANS configures the modem to answer calls. While in ANS mode the Model 100 transmits code at 2225/2025 Hz and receives it at 1270/1070 Hz. When the ANS/ORIG switch is set to ORIG the Model 100 dials to establish communications, transmitting data at 1270/1070 Hz and receiving it at 2225/2025 Hz. Normally you will use the computer in ORIG mode to dial up services such as Compuserve or the Source.

In Telcom Terminal mode you may also toggle between full and half duplex by pressing F4, the Full key in Telcom Terminal mode. Half duplex shows you the characters as you key them. Full-duplex shows you the characters you keyed as the computer on the other end receives them. In this way you may detect transmission errors. Most telecommunications is conducted in full-duplex. See Acoustic Coupler, Modem Cable, Telcom.

Modem Cable, Installing Before using the Model 100 built-in modem for telecommunications you need to connect it with a phone line using the direct connect modem cable (RS-26-1410) or the acoustic coupler (RS-26-3805). Although the modem cable will only work with modular phones, it is the preferable device because it supports the automatic dial and automatic log-on functions of Telcom while the acoustic coupler does not.

The direct connect modem cable has a large cylindrical plug on one end and two small square connectors on the other (just like the one used to plug a modular phone into the wall). One of the two small square connectors is attached to the end of a long silver cable. The other small square connector

is attached to the end of a long beige cable. Both cables are attached to the cylindrical plug on their other end. The modem cable comes with a second cylindrical plug, the shorting plug, that mates to the cylindrical plug on the one end of the modem cable.

To hook the computer up to a modular phone using the modem cable, insert the round plug on one end of the modem cable into the connector marked Phone on the rear panel of the Model 100. Now look at the phone. There will be a cord running between the body of the telephone and the wall outlet. Disconnect one end of that cord from the body of the phone and insert the square connector on the end of the silver modem cable into the telephone instead. Now take the end of the telephone cord that you just disconnected from the phone and plug it into the square connector at the end of the beige modem cable. The modem is now connected to the phone line.

To quickly disconnect the computer from the modem cable, simply remove the cylindrical modem cable plug from the phone port on the rear panel of the computer. Then mate the cylindrical shorting plug and the cylindrical modem cable plug and you will be able to use the phone normally without having to disconnect the rest of the modem cable from the phone. You may also use the telephone normally while the computer is hooked up as long as the computer is not using the phone.

Motherboard The main part of a circuit board in an electronic device. It is usually the largest board and has "slots" for smaller boards to plug into. See Board.

MOTOR BASIC Statement. MOTOR statement will turn the cassette on and off automatically from the BASIC program. The formats are:

MOTOR ON

or

MOTOR OFF

The cassette motor is turned on if you enter MOTOR ON. The cassette motor is turned off if you specify MOTOR OFF. See Cassette Tape, Installing.

MOV M,<r> 80C85 Assembly Language Instruction. MOVE to Memory. The contents of register <r> are moved to the memory location whose address is in registers H and L. The addressing mode is register indirect. No flags are set.

MOV <r>,M 80C85 Assembly Language Instruction. MOVE from Memory. The content of the memory location, whose address is in registers H

and L, is moved to register <r>. The addressing mode is register indirect. No flags are set.

MOV <r1>,<r2> 80C85 Assembly Language Instruction. MOVE Register. The content of register <r2> is moved to register <r1>. The addressing mode is register. No flags are set.

Move Data Down the Screen The PgDn key scrolls the screen down 25 lines and places the cursor in the home position of the screen. This function is common among full-screen editors and word processors. Although a recommended value, this key may be defined differently by various application programs. See Control Keys.

Move Data Up the Screen In TEXT and BASIC EDIT modes, pressing the SHIFT and cursor down (↓) keys together moves the cursor to the bottom of the screen in the current column. If the cursor is already at the bottom of the screen when you do this, the next seven lines scroll up onto the screen from below and while the current screen contents scroll up and off the screen.

To move to the end of the file, press the CTRL and cursor down (↓) keys together.

Moving the Cursor The cursor is the underline symbol (or, with some programs, a different symbol) which appears on the screen to let you know where an action, such as entering, deleting, or inserting a character, will take place. The cursor keys move the cursor basically the same way in all four application programs, BASIC, and on the Main Menu.

The cursor is primarily moved by the cursor movement keys. The Model 100 has four half-size cursor movement keys located on the right end of a row of special keys, directly above the top row of the standard keyboard. The keys are labeled, left to right, ←, →, ↑, and ↓. Normally, if you press any one of these cursor keys once, the cursor moves one character space in that keys' designated direction or if the key is held down, the instruction will automatically repeat. In BASIC Entry mode, SCHEDL and ADDRSS programs, and in Telcom entry mode the ↑ and ↓ keys have no effect because cursor movement is limited to the current line prior to its being entered into BASIC. In this instance, backspacing erases any character the cursor backspaces over. Once the current line is entered it cannot be changed. In BASIC EDIT mode, TEXT, Main Menu, and most host systems in Telcom Terminal mode, the cursor can be moved in all four directions without erasing any of the characters it passes over.

In TEXT and BASIC EDIT, either the SHIFT or CTRL

key must be pressed at the same time as the cursor key. SHIFT and the cursor right key (→) moves the cursor to the first character of the next word to the right of the same line. Pressing SHIFT and the cursor left key (←) moves the cursor to the first character of each word to the left. Pressing CTRL and a cursor right (→) or left (←) key at the same time moves the cursor to the right or left end of the current line. Remember, these key combinations are only valid in TEXT and BASIC EDIT modes. See Move Data Up the Screen, Move Data Down the Screen.

MPU MicroProcessor Unit. The chip containing the processor. A Z80 for the Model 100. This is a CMOS revision of the 8085 microprocessor.

MSB Most Significant Bit. The format used for address storage by the 80C85 and many other microprocessors. The four hexadecimal digits required to address a 64K memory are stored two per byte, but the order of the two pairs is reversed. Thus FEE0 is stored as E0 followed by FE.

MSTRAP* Install the MSTRAP in minutes to provide an over-the-shoulder or hand-carry handle for the Model 100. Guaranteed for life. The Donald Stephens Company.

Multiplan* After VisiCalc's successful release in 1978, the microcomputer was recognized as a viable tool for small businesses. The electronic spreadsheet did not require much programming savvy or skills, could be mastered fairly simply, and provided sophisticated results when applied to the pencil gnawing repetitiveness of daily bookkeeping and financial planning. Soon the market was glutted with VisiCalc imitators. Multiplan is one of the latest and more successful attempts to go VisiCalc one better.

While SuperCalc* tried to out-distance the original in speed, Multiplan's main thrust in the competition is that it is easy to use. VisiCalc uses a row of single letters spread across the top of the screen as a command line, and is at times too cryptic to be attractive to users who are uncertain of their programming expertise. Multiplan has concentrated on simplifying the screen to make the spreadsheet easier and more understandable.

Multiplan does this by using numbers which are easy to remember for both the row and column coordinates of its grid, unlike VisiCalc, which uses letters across the top of the screen. Thus, the first cell in Multiplan, the upper left corner, is referenced as R1C1, for Row 1 Column 1, rather than the A1 of VisiCalc.

Multiplan's also has two rows of command words at the bottom of the screen replacing VisiCalc's line of single letters. The Multiplan screen also displays the current cursor position, amount of memory remaining, a message line, and the name of the worksheet.

Multiplan's commands, Blank, Delete, Copy, Move, and Format, parallel VisiCalc's. Multiplan has seven new commands: Lock, Sort, and Name, which are not found in VisiCalc. Likewise, Multiplan provides the same functions as a spreadsheet, adding 12 more of its own. Multiple use of windows to view different parts of a spreadsheet at one time, has prompted one reviewer to tout Multiplan as a "highly analytical engine" (*PC World*, Vol. 1, No. 2, p. 126).

Creating a spreadsheet involves placing numeric values in the different cells and relating them to other cells, using mathematical formulas. For example, cells R1C1, R1C2, and R1C3 each represent a value and can be added together in the formula R1C1+R1C2+R1C3. The total of this formula appears in a fourth cell. Creating a formula by adding three cells together can become awkward if the cell coordinates are difficult to remember.

Multiplan simplifies this by allowing you to give cells meaningful names and use these names in your formulas. For example, one cell can be given the name SALES for a sales statistic; another cell can be named COSTS. These could then be used in a formula of SALES-COSTS; the sum of these could be placed in a cell called PROFITS. This feature alone represents a significant enhancement in spreadsheet use. Microsoft Corporation.

Multiple BASIC Statements on One Line A colon (:) at the end of one statement allows you to enter another statement on the same line without giving a new line number; the line number at the left refers to all statements on the line. One line may wrap onto several screen lines if you don't press ENTER. An apostrophe (') allows a comment or remark to be added after a statement with or without a separating colon.

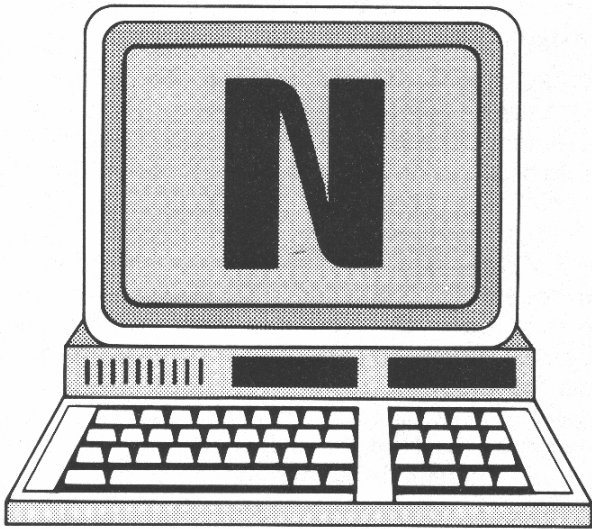
Multiplex Use the same component for several distinct functions, either sequentially or simultaneously. The address lines of an MPU are often used as the data lines at another point in time—an example of multiplexing.

Multi-Tasking Running two or more tasks concurrently on a single computer. The MPU actually works on only one task at a time, but it switches back and forth so rapidly as to create the impression that all tasks are being processed concurrently.

Music BASIC. See SOUND.

MVIM,<data> 80C85 Assembly Language Instruction. Move to memory immediate. The data contained in byte 2 of the instruction is moved to the memory location whose address is in register H and L. The addressing mode is immediate register or register indirect. No flags are set.

MVI <r>,<data> 80C85 Assembly Language Instruction. Move Immediate. The data contained in byte 2 of the instruction is moved to register <r>. The addressing mode is immediate. No flags are set.



N Codes. ASCII 78, HEX 4E. n—ASCII 110, HEX 6E.

NAK Negative AcKnowledge.

NAME...AS BASIC. Changes the name of an existing RAM file. The format is:

NAME "<filename1>" AS "<filename2>"

<filename1> is the name of the RAM file that you want to change.

<filename2> is the new name you will give to the RAM file.

The file declared by <filename1> must be currently in RAM and the file declared by <filename2> cannot be in RAM. The two letter extension of the RAM files must be used and may not be changed in renaming the files. File names may also optionally be specified as "RAM: <filename>" for greater clarity.

Names, Reserved See BASIC Reserved Words.

Names, Variable Rules for Variables in BASIC. See Variable Names.

NAND NOT-AND Logical.

NAND Gate Hardware device using the NOT-AND logical function.

Native Code Machine language.

Native Compiler Compiles programs for the processor on which it runs.

Natural Logarithm BASIC. See LOG.

NBS National Bureau of Standards.

NC No Connection.

NDAC Not Data ACcepted.

NE Not Equal. The sign for not equal is \neq .

NEC Operating System The simple, built-in operating system allows you to perform certain operations from the Main Menu. The functions include: KILL (erase a file), NAME (rename a file), LIST (send a file to the printer through the Centronics port), LOAD (load a file from cassette), SAVE (save a file to cassette), BANK (switch memory banks), SETIPL (create an auto start file), and CLRIPL (eliminate an auto start file). These functions are assigned to the function keys when on the Main Menu.

A program may be loaded from the Main Menu by placing the cursor on the file name. The selected file will be shown in reverse video. Pressing RETURN loads the file, and, if it is a BASIC program, starts execution.

NEC PC-8201A A counterpart of the TRS-80 Model 100, the NEC PC-8201A portable briefcase computer is manufactured by the same company. Kyocera Ltd., a Japanese company, makes each of the computers to the individual specifications of NEC and Tandy. Both machines are very similar but have some important differences.

The unexpanded Model 100 comes with 8K of RAM, while the PC-8201A starts out with 16K. The PC-8201A can be configured with a maximum of 96K of user memory. The Model 100 is limited to 32K when fully expanded. Both machines have 32K of ROM.

The heart of both machines is a battery powered CMOS (80C85) microprocessor. Both models have the same type of LCD display, RS-232 serial port, parallel printer interface, bar-code reader jack, cassette interface, and a built-in clock/calender. The PC-8201A does not have a built-in modem. It does have a CRT adapter, a floppy disk drive interface, and an expansion bus to accommodate plug-in RAM cartridges.

Though similar, the full-size QWERTY keyboards are also different in several ways. The PC-8201A does not have a numeric keypad, foreign character capability, or built-in graphics as does the Model 100. It has five function keys which may be programmed for ten functions by using the SHIFT key. The cursor control keys are arranged in a unique, easy-to-use pattern.

The built-in programs that come with each unit include TEXT, a simple word processor; TELCOM, a data communications program; and variations of Microsoft BASIC. NEC also comes with a cassette tape containing twenty-three application programs.

Many third party software companies have started developing application software for the Model 100

NEC PC-8201A—Hardware Configuration • NEC PC-8201A—Memory Scheme

(Tandy released the Model 100 several months before NEC's machine was released). Some of these developers have started or plan to provide the same programs for the NEC.

NEC PC-8201A—Hardware Configuration The PC-8201A is 11½ by 8¼ by 2½ inches and weighs 3¾ pounds. It has an eight-line, forty-column liquid crystal display (64 by 240 pixels) and a full-size, QWERTY keyboard. Five of the seven interface ports include an RS-232 serial port, a parallel Centronics-comparable printer port, a Hewlett-Packard bar-code reader port, and a cassette interface port. All of these function identically to the Model 100. Additionally, the PC-8201A features two SIO ports for hook-up of an external CRT and a floppy disk drive. It also has a forty-eight-pin expansion bus to utilize the RAM cartridges. A 5½ octave range of sound can be produced. A clock/calender is built-in and can be accessed through BASIC. An optional AC power adapter is also available.

NEC PC-8201A—Keyboard The sixty-seven key, IBM Selectric-style keyboard features full-travel keys with upper and lower case characters. The PC-8201A has a distinctive triangular, “north-south-east-west” cursor control key configuration. Five function keys allow ten programmable functions through use of the SHIFT key. The value of these function keys is determined by the software running and can be user-defined in a BASIC program. The values assigned to each key are displayed at the bottom of the LCD and change to the alternate values when the SHIFT key is pressed. When in BASIC, the five function keys are labelled LOAD, SAVE, FILES, LIST, and RUN, but in TEXT program mode, FIND, FIND-NEXT, SELECT, CUT, and COPY. Two of the “home” keys, the F and J, have tiny raised dots to aid in typing. The PC-8201A keys automatically repeat if held down for a moment.

The PC-8201A has a STOP key which halts program execution just like Control-C. Other keys include ESC (escape), TAB (tabulate), CTRL (control), CAPS (cap-lock), SHIFT (upper to lower case), GRPH (graphics), DEL (delete), BS (backspace), INS (insert), and PAST (paste). The PAST key allows retrieval of a previously identified piece of text for insertion into the current text. The GRPH key allows access to ninety-three characters, ninety of which are user-defined with the CHRDEF program. Another thirty-two user-defined characters are available through use of the CHR\$ BASIC command. The DEL key deletes characters under the cursor, while the BS key moves the cursor to the left. The INS key inserts a character above the cursor, moving the character

currently above the cursor and those that follow, to the right.

NEC PC-8201A—Memory Scheme The NEC PC-8201A can be upgraded to a maximum of 96K, in 8K increments. The 96K of RAM is organized in three banks of 32K each. Only two of the three banks can be active at one time. Each memory bank can contain up to twenty-one files.

One of these banks is in the form of a RAM disk cartridge which is plugged into the expansion bus on the side of the NEC. The RAM cartridge is powered by its own battery, and maintains stored data for about six months, or indefinitely, if attached to its AC adapter. This bank, as well as the second internal bank, has write-protect switches to prevent accidentally overwriting the contents.

The RAM cartridge is plugged into the side of the computer, and a “cold boot” is performed by pressing the SHIFT key, CTRL key, and function key five simultaneously. This will initialize the RAM cartridge and must only be done the first time the cartridge is used. All files currently in memory should be stored on another cartridge or on cassette before this procedure is attempted. Thereafter, the cartridge is simply plugged in and accessed by activating that bank.

Four alkaline batteries power the PC-8201A for up to 18 hours (on a 16K RAM machine). An optional AC adapter may be used. The battery discharge prevention switch is used to continue the power supply after the machine is turned off. All files in RAM are erased if this switch is turned off. An internal nicad battery, which powers the clock and the internal RAM, lasts up to twenty-six days (on a 16K RAM machine) and can be recharged by using the alkaline batteries or an optional AC adapter. This power source saves RAM, even if the main batteries go dead.

When the batteries need to be changed, as indicated by the LED on the face of the machine, the AC adapter is used temporarily to maintain the files. The batteries are housed in a removable battery pack. The automatic power off feature turns the computer off, if not in BASIC, and no key is pressed for ten minutes, thus saving the battery power. This feature may be disabled or changed from BASIC.

The PC-8201A also features 64K ROM packs which must be installed at a service center. NEC plans to offer additional application programs in these ROM packs.

The LCD can display both upper and lower case characters. The display can be controlled by slop-

ing the screen and lightening or darkening the contrast. Reportedly, the LCD should be protected from extreme cold because it can freeze.

The RS-232 serial port can be used to connect a printer or a modem. The port configuration can be varied by using the built-in communication software. A direct connect modem from NEC or the battery-powered NEC thermal printer can be connected through this port. The parallel printer interface will run most Centronics-compatible printers with the special cable available from NEC. There are several third-party bar-code readers available which will run with the PC-8201A. NEC has its own cassette recorder, the PC-2081, but the machine will work with other recorders as well.

NEC PC-8201A—Software The PC-8201A comes with three built-in programs, BASIC, TEXT, TELCOM, as well as a simple operating system. There is also a cassette tape which contains twenty-three application programs. While the similarities between the Model 100 and the NEC PC-8201A are great, the differences are sufficient to cause most software for one machine to fail to run on the other. Be sure any software you purchase is clearly marked as compatible with your machine. Do not assume you can use Model 100 software on the NEC or vice versa. Most software will eventually be available in two versions, one for each machine.

BASIC has a text editing mode which can be used to create BASIC programs. The PC-8201A allows you to switch between BASIC editing mode and the TEXT word processing mode when developing a BASIC program.

NEC PC-8201A BASIC See BASIC, NEC PC-8201A.

NEC PC-8201A Cassette Software The PC-8201A comes with a cassette tape loaded with twenty-three application programs:

CALC makes the PC-8201A act like a four function (addition, subtraction, multiplication, division) calculator which can handle up to 100 calculations at a time.

TXFORM is a text formatter that will set margins and tabs, define page length, number pages, and allow for user-defined titles. Designed to be used with TEXT.

INVEST is an investment tracking program which can update the portfolio file, examine selected investments, and print summary reports of the portfolio to screen or printer.

FCAST calculates the linear forecast of values for a specific period. Displays the results graphically on the screen.

SCHDL functions like the SCHEDL program that comes with the Model 100, except that it features an alarm that will beep at any desired day and time. CHRDEF allows thirty-five graphic or foreign characters to be defined by the user and stored for future use.

BACKUP,XFILES is used for file manipulation between the PC-8201A and disk emulator.

BA, BACUT is used to access the three memory banks.

TERM defines and stores data communication parameters.

JAN, NWH, COD, BCR, DEMO are used with the bar-code reader.

MUSIC stores and plays back musical creations, using the keyboard as a piano.

TANK is a game in which you are the captain of a tank and must destroy your enemy before he destroys you.

SNAKE is a game where you use rocks piled in the path to avoid being bitten by the snake.

NEC PC-8201A Software and Hardware Many of the software and hardware houses which are developing products for the Model 100 are now doing the same for the NEC PC-8201A.

The following products are currently available for the NEC PC-8201A:

Praire Power—system recharger from Bluestem Productions.

Bar Code Reader—from BT Enterprises.

Business programs from Traveling Software, including: The Traveling Time Manager, The Traveling Sales Manager, The Traveling Expense Manager, The Traveling Accountant, The Traveling Project Manager, The Traveling Tax Manager, and The Traveling Communicator.

Business analysis programs from American Micro Products Inc. (AMPI), including: Equation Solver, Histogram and Plot, Income Property Analysis, Portfolio Analysis, RPN Calculator, Statistical Curve Fitting, and T Plan/N Plan Spreadsheet.

Forth-79 is a Forth language version from American Micro Products, Inc.

ViewDex is an indexed filing system from Quickview Systems, Inc.

8K memory upgrade modules can be used in place of the more expensive ones available from NEC. Purple Computing.

NEC PC-8201A TELCOM This is the telecommunication program which is provided in the ROM of both the Model 100 and the PC-8201A. It controls

communication with other devices through the RS-232 port. Since the PC-8201A does not have a built-in modem, this program may be used when using an external modem, to vary the baud rate from 300 baud to 19200 baud. (The Model 100 is limited to 300 baud when using the internal modem.) This program can also be used to set the bits-per-character, parity, stop bits, handshaking, and full or half duplex.

NEC TEXT The built-in word processing program functions just like the Model 100 version. This program can be used to write BASIC programs or for creating text files. It can also be used to create an IPL (initial program load) file, which contains a series of commands to be used at start-up time.

The TEXT program always operates in "insert" mode. The cursor controls move one character or line at a time, unless used with the SHIFT key. Then they move right or left a word at a time or up and down a screen at a time. The cut and paste functions allow a section of text to be cut and copied or moved to a "paste buffer." This section can then be pasted into a new location. The Search function finds occurrences of a string of characters up to 24 long, while ignoring the case of the letters. The NEXT command continues the search for the same string (this command is not available on the Model 100). There is also an auto word-wrap feature.

Negative Logic When a normally true-state voltage in the computer system represents a logically false state. A normally false-state voltage in the system represents a logically true state.

NEQ Stands for Not EQal to (\neq).

Nested Subroutine A subroutine within a subroutine. The inner subroutine (nested) is completed first, and the outer subroutine is completed after the nested subroutine has finished. You may also nest FOR...NEXT loops and IF...THEN...ELSE statements.

Network Linking computers to form an interconnected system, or a network.

NEW BASIC Command. Deletes the program currently in computer memory and clears all variables. The format is:

NEW

NEW is generally used to erase the current program in BASIC and clear all memory before entering a new program. It sets numeric variables to zero (0) and string variables to null (" "). BASIC always returns to command level after NEW is executed. NEW closes all files. If you don't use NEW to clear

BASIC memory before entering a new program, you may get a mix of new and old program lines.

New File Create in RAM. See RAM File, Create.

NEXT BASIC. End of FOR...NEXT loop. See FOR...NEXT.

Nibble Half a byte, or 4 bits.

Nicad Battery The Model 100 contains a built-in rechargeable nicad battery to back up RAM memory if the AA alkaline batteries wear out or wall current is disconnected. During a power failure, batteries supply auxiliary power to the microprocessor, so volatile information is not lost.

Noise Random, unknown interference signals on a system communications line.

Non-Destructive Read-Out Memory contents are not erased or displaced when other data is read.

Non-Text Files See Editors, Edit.

NOP 80C85 Assembly Language Instruction. No OP. No operation is performed. The registers and flags are not affected.

NOP or NOOP No OPERATION. An instruction which forces a delay of an execution cycle but does not change the contents of the registers or the status flags.

NOR Condition of being NOT-OR

Normal Size Print See Type Formats.

NOT Negative operation which changes every 1 in a byte to 0 and every 0 in a byte to 1. Also used as a logical operator to reverse the outcome of a conditional test in many languages (If x NOT = 1).

NOT-AND Logical Operator A NOT-AND B is a synonym for NOT (A AND B).

Notch See Write-Protected Diskette.

NOTE.DO File See SCHEDL Mode.

NSEC NanoSECond. One nanosecond equals one billionth of a second.

Null Detector A circuit which registers when current is not flowing or when voltage is not present.

Null String or File A string or file with no contents. Variables remain null until assigned values by BASIC. After entering CLEAR, all variable contents are also null.

NUM Key Converts a segment of the standard alpha keyboard to a trapezoidal numeric keypad that also uses the standard numeric keys 7, 8, and 9.

Number, Largest Line • Numeric Variables, Converting

Pressing the NUM key twice returns the keys to their previous alpha values.

Numeric Keypad Key Values	
NUM OF	NUM ON
M	0
J	1
K	2
L	3
U	4
I	5
O	6
7	7
8	8
9	9

Number, Largest Line BASIC. The largest line number for a BASIC program is 65529.

Number Crunching The performance of complex arithmetic operations and computations.

Numeric Pad See NUM Key.

Numeric Value of String BASIC. See VAL.

Numeric Variable, Double-Precision BASIC. The default variable type in Model 100 BASIC is numeric double-precision (numbers with decimal fractions up to fourteen significant digits). Otherwise, numeric double-precision variable names end in # or start with a letter specified in a DEFDBL statement.

Double-precision, floating-point numeric variable names must start with a letter and can have any number of characters, although only the first two are significant. They must not be any reserved words, such as IF, ON, THEN, GOTO, etc., a reserved word followed by a type declaration character (\$, %, !, #), or a reserved word embedded in the variable name as AND is embedded in HAND. See Reserved Words, Uses and Restrictions of, and Variable Names.

Numeric Variable, Integer BASIC. Numeric whole number variables hold integers with values ranging from -32768 to +32767, end in %, or start with a letter specified in a DEFINT statement.

Integer numeric variable names must start with a letter and can have any number of characters, although only the first two are significant. They must not be any reserved words, such as IF, ON,

THEN, GOTO, etc., a reserved word followed by a type declaration character (\$, %, !, #), or a reserved word embedded in a variable name as LEN is embedded in LEND. See Reserved Words, Uses and Restrictions of, and Variable Names.

Numeric Variable, Single-Precision BASIC. Numeric single-precision variables hold numbers with decimal fractions and up to six significant digits; variable names end in ! or start with a letter specified in a DEFSNG statement.

Single-precision numeric variable names must start with a letter and may have any number of characters although only the first two are significant. They must not be any reserved words, such as IF, ON, THEN, GOTO, etc., a reserved word followed by a type declaration character (\$, %, !, #), or a reserved word embedded in a variable name as AND is embedded in HAND. See Reserved Words, Uses and Restrictions of, and Variable Names.

Numeric Variables, Converting Instructions and examples for various types of conversions follow: Converting from Double Precision to Integer Values in BASIC: This conversion may be easily accomplished by assigning a double-precision value, variable or constant, to an integer variable. BASIC truncates all digits to the right of the decimal point. The resulting number must be in the valid range for integer values, anywhere from -32768 to 32768. If the value is out of range, the message "?OV Error," the overflow error message, appears. A sample conversion is:

```
OK
d%=34.984532871234
OK
Print d%
34
OK
```

Converting from Double- to Single-Precision Numbers in BASIC: This conversion may be accomplished in BASIC by assigning a double-precision number, constant or variable, to a single-precision variable. BASIC rounds the digits to the right of the decimal so that the number is reduced from fourteen to six significant digits. It uses a rounding method known as 4/5 rounding which rounds down numbers from one to four and rounds up numbers from five to nine. For example:

```
OK
n!=56.467853987637
OK
Print n!
56.4679
OK
```

Numeric Variables, Converting

Converting from Integer to Double-Precision Numbers in BASIC: This conversion may be easily accomplished by assigning an integer value, variable, or constant to a double-precision variable. Remember that the default variable type in BASIC is double-precision numeric. BASIC adds the number of zero digits to the right of a decimal point necessary to equal fourteen significant digits in all. For example:

```
OK
d#=34
OK
Print d#
34.0000000000000
OK
```

When performing mathematical operations using variable or constant values of different numeric types, BASIC temporarily converts all numbers to double-precision for the duration of the calculation. They are not changed in their original memory locations and when stored back into variables take on the type indicated by the variable.

Converting from Integer to Single-Precision Numbers in BASIC: This conversion may be easily accomplished by assigning an integer value, variable, or constant to a single-precision variable. BASIC adds the number of zero digits in all. For example:

```
OK
d!=34
OK
Print d!
34.0000
OK
```

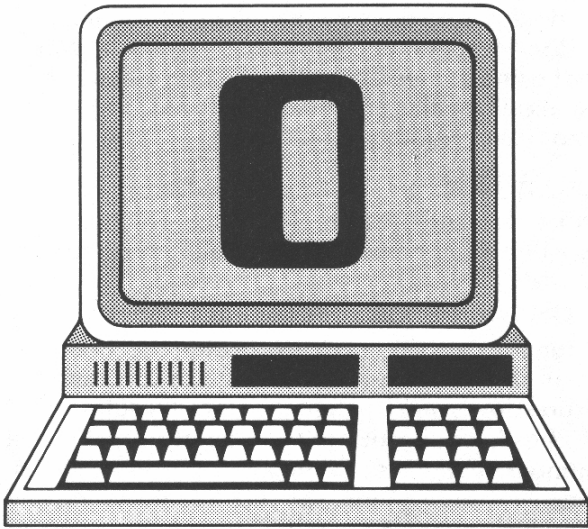
Converting from Single-Precision to Integer Values in BASIC: This conversion may be easily accomplished by assigning a single-precision value, variable, or constant to an integer variable. BASIC truncates all digits to the right of the decimal point. The resulting number must be in the valid range for integer values—from -32768 to 32768. If the value is out of range, the message “:OV Error,” the overflow error message, appears. A sample conversion is:

```
OK
d%=34.1234
OK
Print d%
34
OK
```

Converting from Single- to Double-Precision Numbers in BASIC: This conversion may easily be accomplished by assigning a single-precision number, constant or variable, to a double-precision variable. Remember that the default variable type

in BASIC is double-precision numeric. BASIC adds zeros to the right of the decimal point until the number is increased from six to fourteen significant digits. For example:

```
OK
d#=36.8907
OK
Print d#
36.8907000000000
OK
```



- Codes. ASCII 79, HEX 4F. o—ASCII 111, HEX 6F.
- Output or Overflow.

Object Code The code produced by a compiler or Assembler program. The object code can be executed directly by the computer or may need further translation through a linkage. Model 100 object code is stored in .CO (Command) files.

Object Program Machine language program resulting from translation of a source program.

Object Programs Programs running directly on the Model 100 are Machine language programs using the numeric instruction code of the 80C85 microcomputer chip. The source program contains statements in languages such as FORTRAN, COBOL, or BASIC. These are translated by a compiler program, producing an object program.

The object program contains the Machine language instructions for the 80C85, which correspond to the instructions of the original source program. BASIC programs work with a BASIC compiler. Regular BASIC works in a slightly different way. It acts as an interpreter, using your BASIC program as a guide to what it should do. Since it must re-interpret your source program each time you run it, interpreted BASIC can be as much as 100 times slower than compiled BASIC.

When you write a BASIC program, you have produced a source program in text form. The BASIC interpreter, itself a Machine language program, uses your source program as data—a source of instructions or commands to control its execution. If you decide to speed up a BASIC source program by compiling it into a faster Machine language pro-

gram, you need a BASIC compiler. This program isn't supplied with the Model 100, but is available from software houses. You can run your BASIC compiler, giving it as input the source program (.BA) that you wrote. The compiler translates your .BA source program into an executable Machine language program. The resulting .CO program is no longer a BASIC program; it is a Machine language program which is based on (or is a translation of) your BASIC source program.

Odd Parity Parity is a means of detecting the loss of bits in a file transfer. All bit arrays are set to an even or odd number of bits, and these are checked after the file transfer is complete. Odd parity sets the numbers of bits sent per byte to an odd number. See Parity.

Off-Line Off-line means that a device or medium is not connected to the computer system. Contrast with on-line, which means that a device or medium is inside or connected.

A cassette is off-line when it is in its holder and on-line when it is in a recorder that is turned on and connected to the computer. A printer is on-line if it is attached, on, and ready to print. The printer is off-line if it is disconnected, off, or switched to its "off-line" state by an on-line/off-line switch. A user is on-line if his hands are on the keyboard and off-line if he is at the 7-11, buying milk. See also Memory and Input/Output.

Offset Address The smaller part of a base register/offset pair. The data to be processed is located at: <effective address><contents of base register> + <offset>. Offset Addresses are not used by Model 100 80C85.

OK BASIC. A prompt letting you know the program is waiting for you to type in information. You can enter a BASIC command or statement whenever you see the OK prompt.

? is the BASIC prompt signalling you to enter data to answer a question. In this case, it is a good idea to include a prompt in your BASIC program such as "Enter check amount". This prompt helps the operator (or you) know exactly what should be entered in response to the ? prompt. The format is:

```
100 INPUT "ENTER CHECK AMOUNT",
    CHECKAMOUNT
```

This will give the operator using this program this prompt on the screen:

```
ENTER CHECK AMOUNT ?
```

ON COM GOSUB BASIC Statement. Specifies a subroutine line number in which BASIC can trap

ON...ERROR GOTO • ON KEY GOSUB

program execution if information is received via the RS-232C communications interface. The format is:

ON COM GOSUB <linenum>

<linenum> is the number of the line which begins the trap subroutine. If you set <linenum> to zero (0), the trapping is disabled.

COM ON must be executed to start trapping by the ON COM GOSUB statement. Before each statement is executed, BASIC checks to see if the RS-232C communications device has received any data. If so, a GOSUB is executed to the <linenum> specified by the ON COM GOSUB statement. A RETURN at the end of the subroutine returns execution to the line where the interrupt occurred after the subroutine has been executed.

COM OFF stops the trapping as specified by the ON COM GOSUB statement. After a COM OFF has been executed, RS-232C activity has no effect on program execution.

COM STOP can begin after an ON COM GOSUB so that no trapping takes place for the RS-232C interface. However, characters that are received via the RS-232C are remembered so that a trap takes place immediately upon the next execution of COM ON. Trapping only occurs when four conditions are met: BASIC is executing a program; the ON COM GOSUB statement has been read; the ON COM GOSUB has been enabled by a COM ON statement; characters are received via the RS-232C interface.

Ending the ON COM GOSUB subroutine with a RETURN continues program execution at the place where the interrupt occurred. You can also use a RETURN <linenum> to return from the trap subroutine to the program at a fixed <linenum> other than the one executing at the time of the interrupt.

ON...ERROR GOTO Allows an error trapping interrupt and defines the first line of an error handling subroutine. The format is:

ON ERROR GOTO <linenum>

<linenum> is the line number of the first line of the error handling subroutine. If the line number does not exist within the current program an "UNDEFINED LINE NUMBER ERROR" message appears.

After error trapping by the ON ERROR GOTO statement has begun, all errors found, including direct mode errors, cause the error handling subroutine, beginning at <linenum>, to begin.

The error handling subroutine can end by terminating program execution with an END or STOP statement, or by resuming program execution.

To do this, stop the error handling subroutine with a RESUME statement. It returns program execution to the line immediately after the one that was executing when the error trap occurred. See BASIC Error Code Numbers and ERROR.

ON...GOSUB BASIC. Finds the value of a numeric expression and uses it to determine which subroutine line number execution should branch to. The format is:

ON <numex> GOSUB <linnum>, <linnum>...

<numex> is a numeric expression that evaluates to an integer value. If the value of the numeric expression is one (1), then program execution goes to the first subroutine line number on the line number list. If the <numex> evaluates to two (2), then execution goes to the second subroutine line number on the list, and so on. If the value of <numex> exceeds the number of elements on the line number list, or if it evaluates to zero (0), then execution of the program continues without branching to any subroutine.

<linnum> this is the starting line number of a subroutine, elsewhere in the BASIC program, to which execution will jump if the <numex> indicates it should. See above. A RETURN is used at the end of the subroutine, that begins at each <linnum>, to restore a program execution to the line following the ON...GOSUB that called the subroutine. See ON...GOTO.

ON...GOTO BASIC. Finds the value of a numeric expression and uses it to determine which line number, on its line number list, execution should go to. The format is:

ON <numex> GOTO <linnum>, <linnum>...

<numex> is a numeric expression that evaluates to an integer. If the value of the numeric expression is one (1), program execution goes to the first line number on the line number list. If the numex is two (2), execution goes to the second line number on the list, and so on. If the value of <numex> exceeds the number of elements on the line number list, or if it evaluates to zero (0), execution of the program continues without branching.

<linnum> is the number of a line, elsewhere in the BASIC program, to which execution jumps if the <numex> indicates it should.

A return is used at the end of the subroutine that begins with <linnum> to restore program execution to the line following the ON...GOTO that called the subroutine.

ON KEY GOSUB BASIC Statement. Specifies the line ON KEY(n) numbers to which BASIC traps if

the specified function keys are pressed. The format is:

ON KEY GOSUB <linnum>,<linnum>...

<linnum> is the number of the first line of the trap subroutine for the specified key. The first subroutine (line number) listed is assigned to F1, the second subroutine <linnum> to F2, etc.

KEY ON must first be executed to activate the subroutine assignments made in the ON KEY statement. Then, if a non-negative line number is declared in the ON KEY statement, BASIC checks every time the program starts a new statement to see if a function key was pressed. If a function key was pressed, a GOSUB to the specified subroutine line number is performed. Pressing an undefined key does not cause an interrupt.

KEY OFF means no trapping takes place for the function keys. If a function key is pressed, the event is not acted upon or remembered.

KEY STOP means no trapping takes place for the function keys. subroutine.

Event trapping does not take place when BASIC is not executing a program. An error trap (caused by an ON ERROR statement) disables all trapping including ERROR, COM, MDM, and KEY.

After the ON KEY subroutine has been executed you may use RETURN <linnum> if you want to go back to a specific line number in the BASIC program. Since any other GOSUBs, WHILEs, or FORs that were active at the time of the trap remain active, you must be careful using this form of return.

RETURN alone returns program execution to the next program line after the one that was executing when the ON KEY trap occurred.

ON MDM GOSUB BASIC Statement. Specifies a subroutine line number for BASIC to trap program execution to, if data is received through the built-in modem. The format is:

ON MDM GOSUB <linenum>

<linenum> is the number of the line which begins the trap subroutine. If you set <linenum> to zero (0), trapping of communications activity through the built-in modem is disabled.

MDM ON must be executed to activate trapping by the ON MDM GOSUB statement. Before each statement is executed, BASIC checks to see if the built-in modem has received any data. If so, a GOSUB is executed to the <linenum> specified by the ON MDM GOSUB statement. A RETURN at the end of the subroutine returns program execution to the line that was executing when the interrupt occurred.

MDM OFF stops trapping of the modem interrupt. After a MDM OFF has been executed, modem activity has no effect on program execution.

MDM STOP can be executed after an ON MDM GOSUB statement, so no trapping takes place for activity through the modem. However, characters that are received through the modem after a MDM STOP, are remembered so that a trap takes place immediately upon the next execution of MDM ON.

Trapping occurs only when four conditions are met: BASIC is executing a program; the ON MDM GOSUB statement has been executed; the ON MDM GOSUB has been enabled by a MDM ON statement; data has been received through the built-in modem.

Ending the ON MDM GOSUB with a RETURN continues program execution at the place where the interrupt occurred. Alternatively you may use a RETURN <linenum> if you wish to return from the trap subroutine to the program at a fixed <linenum> other than the one that was executing at the time of the interrupt.

ON TIME\$=...GOSUB BASIC. Defines an interrupt subroutine beginning at a specified line number which is executed when the system clock time equals the given value of the time variable. The format is:

ON TIME\$= "<hh>:<mm>:<ss>"
GOSUB <linnum>

"<hh>:<mm>:<ss>" is the time you are setting the interrupt on the system clock for. It gives hour, minute, and second, each in a two digit format, separated by colons (:). This variable should be twenty-four hour clock time and include both digits for each interval, even if that means including leading zeros.

The format of TIME\$ is also twenty-four hour time in two digits and separated by colons. It may be set and displayed by the user, but is maintained and updated by the system.

<linnum> is the line number of the subroutine program execution jumps to when the time variable coincides with the system time.

Even though an ON TIME\$ has been defined and read by the program, no trapping occurs unless the TIME\$ interrupt has been enabled using a TIME\$ ON.77 See TIME\$ ON BASIC.

One Line, Multiple BASIC Statements on See Multiple BASIC Statements On One Line.

On-Line On-Line means that a device or media is connected to the computer system. Contrast this

Open • OPEN

with off-line, which means that a device or medium is disconnected from the computer system.

A cassette is off-line when it is in a box, and on-line when it is in a cassette recorder which is plugged in and attached to the computer with cassette cables. A printer is on-line if it is attached, powered on, and ready to print. The printer is off-line if it is disconnected, powered off, or switched to its "off-line" state by an on-line/off-line switch. See also Memory and Input/Output.

Open The files of a program must be open in order to run, so that the data contained is available to the running program.

OPEN BASIC Command. Opens a file for input and output from a BASIC program. OPEN accesses any of six devices. It assigns a buffer to the device file it OPENS. It assigns a file number. The device and file can then be referenced by BASIC for input and output by specifying only the file number until you close the file. The formats are:

```
OPEN ["<dev:>"] <filename>
FOR <process> AS <filename>
```

or

```
OPEN "<dev:> <configuration>"
FOR <process> AS <filename>
```

or

```
OPEN "<dev:>
```

<dev:> is the device associated with the input/output file. It is optional, and, if omitted, is assumed to be RAM:. Valid device specifications in an OPEN statement are:

RAM: a RAM storage file. The default value for <dev:>.

CAS: a cassette storage file.

COM: a communications file transmitted via the RS-232C interface.

MDM: a communications file transmitted via the built-in modem.

LPT: a file displayed on the printer. Output only.

LCD: a file displayed on the LCD screen. Output only.

The RAM: and CAS: devices use the first format, which specifies a <filename>.

The COM: and MDM: devices use the second format, which specifies a <configuration>.

The LPT: and LCD: devices use the third format, which needs only the device specification <dev:>.

<filename> is a string of six characters or less that begins with a letter. It is the name that BASIC assigns to the storage files it accesses for I/O to the RAM: and CAS: devices. If the device is RAM:, you

may optionally specify one of the three, two-character, RAM file extensions .DO, .BA, or .CO. If the device is CAS:, no file extension is used.

<configuration> is a five character string in the format <r><w><p><s>, where each character represents one communications parameter to be used in transmitting file I/O to another computer through the COM: and MDM: devices. Values for each communications parameter are:

<r> is the baud rate. Valid values are listed below:

- 1 = 75 baud
- 2 = 110 baud
- 3 = 300 baud
- 4 = 600 baud
- 5 = 1200 baud
- 6 = 2400 baud
- 7 = 4800 baud
- 8 = 9600 baud
- 9 = 19200 baud

If the communications file you are opening is specified as a modem communications file (MDM:), then the <r> communications parameter value should be omitted. The modem operates at 300 baud (m) by default. If the default value is altered, even by specifying default itself, then the built-in modem will be disabled and the RS-232C interface enabled.

<w> represents word length in bits, and may be set to values 6, 7, or 8.

<p> represents parity and may be set to:

- E = even parity
- O = odd parity
- I = ignore parity
- N = no parity

 indicates stop bits. 1 for one stop bit or 2 for two stop bits.

<s> enables and disables XON/XOFF status. E enables it, and D disables it.

<process> specifies the type of input or output you are OPENing the device files for. You may specify INPUT, OUTPUT, or APPEND. Some devices will only allow a limited range of I/O processes:

RAM: INPUT, OUTPUT, and APPEND

CAS: INPUT and OUTPUT

COM: INPUT and OUTPUT

MDM: INPUT and OUTPUT

LPT: OUTPUT only

LCD: OUTPUT only

<file number> is the file number that represents the I/O buffer and device file pair created by the OPEN statement in subsequent BASIC I/O commands. Input or output to or from the file named by the OPEN statement is handled by BASIC com-

mands INPUT#, LINE INPUT#, PRINT#, and PRINT# USING, which refer to the file by <filename> only. The association between device file, buffer, and file number is terminated by the BASIC CLOSE command.

To open more than one file in BASIC, you must change the system value of the MAXFILES variable from the default value of 1 to any number you require up to 15.

Operand Operations are performed upon entities called the operands. For example, LET A=B+C specifies that the operation of addition be performed on the operands B and C.

Operating System The software which manages the hardware and the logical reasoning of the computer. The operating system manages files, schedules processes, and handles any connected devices.

Operation Action of a program resulting in a defined instruction being executed and carried out.

Operation Code Part of the Assembler or Machine language program which specifies an operation to be performed.

Operator See User.

Operator Symbols within a program which specify what operation is to be performed. Examples are * for multiply, - to subtract, or + to add. See Mathematical Hierarchy of Operations Table.

Operators BASIC Math and Relational Operators Tables

BASIC Math Operators Table

Math Operator	Operation Performed
+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Integer Division
^	Exponentiation
MOD	Modulus Arithmetic

BASIC Relational Operators Table

Relational Operator	Function
<	Less than
>	Greater than
=	Equal to
≠ or <>	Not equal to

=> or ≥ Greater than or equal to
=< or ≤ Less than or equal to

Optimization Changing either the software or the hardware of a computer system so that it operates faster or more efficiently.

Options You can choose the communications parameters used by the modem or RS-232C interfaces in Telcom Terminal mode or from BASIC using the SAVE, LOAD, and MERGE commands.

Options from Menu, Choosing A menu is a screen display which lists a number of possible options and asks the user to select one. When a selection is made by keying in an identifying number or letter (or positioning the cursor beside the desired item, using a light pen, etc.), the selected function is performed. This may require either a branch or subroutine call to the code for the function, or the program to carry out this function may be loaded into memory and executed. See Main Menu.

OR OverRun. The status flag of the Universal Synchronous/Asynchronous Receiver/Transmitter (UART) goes to 1 if a new character is written on top of an old one.

ORA M 80C85 Assembly Language Instruction. OR memory. The contents of the memory location, whose address is contained in the H and L registers, is inclusive OR'd with the contents of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

ORA <r> 80C85 Assembly Language Instruction. OR Register. The contents of register <r> is inclusive OR'd with the contents of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared. The addressing mode is register. Z, S, P, CY, and AC flags are set.

ORI <data> 80C85 Assembly Language Instruction. OR Immediate. The content of the second byte of the instruction is inclusive OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared. The addressing mode is immediate. Z, S, P, CY, and AC flags are set.

OS Operating System.

OUT BASIC Statement. Sends a byte value to a designated, CPU output port. The format is:

OUT<prtnum>,<bytval>

<prtnum> is a numeric expression with a value between 0 and 255.

Out • Over-Voltage Protection

<bytval> is a numeric expression for the byte value data to be transmitted to the port. It has a value between 0 and 255.

OUT is used with the INP function.

OUT <port> 80C85 Assembly Language Instruction. Output. The content of register A is placed on the eight bit bi-directional data bus for transmission to the specified port. The addressing mode is direct. No flags are set.

Output BASIC. See PRINT USING, LPRINT USING.

Output Information coming out of a program. It may be going to the LCD, printer, cassette, speaker, etc.

Output WRITE data from memory. See Memory.

OV Overflow. An overflow occurs when the number of digits that result from a mathematical operation exceeds the number of registers available to hold the result.

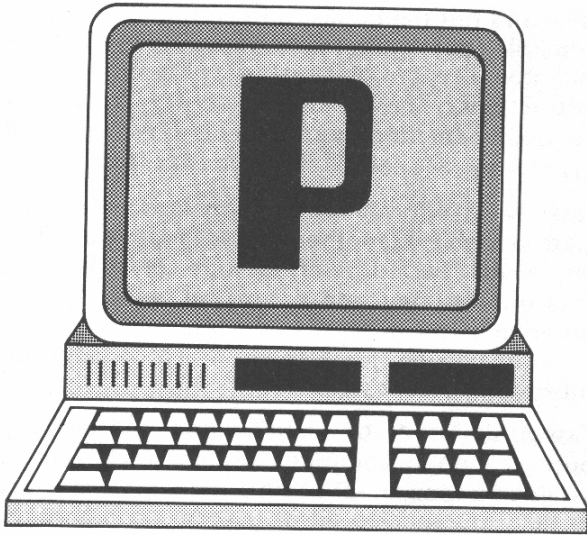
Overflow The sign bit is filled from the next lower bit, and a status flag is set. The result of the arithmetic operation is too large for the specified register.

Overlay Different routines within a program which occupy the same memory location. They are loaded as needed during the operation of a program. Conflict occurs if two routines with different overlays try to use the same location at the same time.

Over-Run Error The previous character in a register hasn't been completely read when a new character is loaded into the register.

Over-Striking The printer returns to a position where a previous character was struck and strikes a new character on top of it producing a combination character. Also called boldfacing if the same characters are struck twice. See Type Formats.

Over-Voltage Protection The computer's circuitry is protected from undesirable increases in the power line voltage.



P Codes. ASCII 80, HEX 50. p—ASCII 112, HEX 70.

P Parity. See Parity.

Package, Software A group of computer programs, possibly including data files and documentation, which performs a function or group of related functions on the computer. These are called applications software when they help the user with some task, such as a word processing package, or an accounting package. These programs are called systems software packages when they facilitate the use of the machinery, such as a database management package, a disk operating system, or a program development package.

Packed Decimal Two or more binary coded decimal digits are present in every byte. The 10 decimal digits 0 through 9 are stored in a 4-bit representation. 1 is coded as 0001, 9 as 1001, etc.

Packet A short set of data with the length of the data stored in the first few bytes. The set is then transferred in a network of devices.

Pad The rectangular base where the wire of a computer chip is connected.

Pad To fill a data field with blanks. See PRINT.

Page Length To set printer page length to <n> lines per page, enter the BASIC statement:

```
LPRINT CHR$(27);"C";CHR$(<n>)
```

<n> may range between 1 and 127 and represents lines per page. See Type Formats.

Page, Memory A logical set of storage used for the management of memory. In a paging system, the memory location is designated by a page number and an address within the page.

Paging To switch from one full screen of information to another, for example, while word processing. You can page either backward or forward.

Parallel Simultaneous handling of processes, transmissions, or storage of data. In most microcomputers, parallel I/O connections have eight wires to carry eight bits of a byte simultaneously (or in parallel). Contrast this with a serial interface, where only one data wire is available. The eight bits of a byte are transmitted one after another (or serially). The I/O device must collect all eight bits and reconstruct them into one eight bit byte. The RS-232C interface and the built-in modem both transform data from the parallel format used internally by the Model 100 to the serial format used for data transmissions between computers, and vice versa. See Printer, How to Hook Up.

Parameter Another name for a definable variable in a program or system which can be used to send information to and from a subroutine or procedure. A parameter may take on various values which are used in subsequent processing by a program. You may optionally specify two parameters to be passed to a Machine language subroutine from BASIC using the CALL statement.

Parity Error detection technique that uses a one-bit indicator at the end of a word. Odd parity sets the parity bit to make the total number of one bits odd. Even parity sets the parity bit to make an even number of one bits.

Parser A routine that analyzes a program instruction to set up its tree structure according to the programming language's syntax.

Pascal A structured, and therefore more predictable, programming language developed by Niklaus Wirth in the late 60s. Although Pascal was originally developed as a teaching language, it has become popular with computer professionals and beginners. Its additional features include "records" (data structure definition) and "sets." All structured languages represent a major improvement for programmers who want to closely predict the behavior of a program. It is much easier to understand the function of an IF...THEN...ELSE structure than to trace GOTOs and statement labels to remote, unpredictable places in a program. Pascal has most of the structures necessary to write a completely structured program (without GOTOs). Some of the dialects of Pascal implement all the necessary structures. In any case, it allows for more complete structuring than even FORTRAN 77 or the (semi) structured dialects of COBOL.

Pascal is strongly typed. The "type" (floating point, integer, set, record) of a variable is traced even through procedure calls. This helps to point out a major class of programming errors. Pascal, like other recent languages, requires the explicit definition of all variables as a guard against introducing a misspelled variable. This is otherwise difficult to spot.

The Pascal standard includes an input/output package. This is important for a language intended to be used in textbooks, but limits the kind of I/O that can be done from Pascal.

Pointers and records let the high-level language programmer use powerful techniques formerly only available to Assembly language programmers for handling complicated data structures such as linked lists and trees. You can handle these structures explicitly in Pascal. In FORTRAN these structures would have to be "simulated" inside an array with the structure diagrammed in documentation or the inside of the programmer's head.

Pascal should soon be available for the Model 100. The standard manual is the *PASCAL: User Manual and Report*, by Kathleen Jensen and Niklaus Wirth. It is suitable for a wide variety of chores, from systems to application programming. This manual makes it easy for beginners to understand the function of a program. See UCSD p-System and Pascal MT+.

Pascal MT+ A version of Pascal marketed by Digital Research, the company which also brings you CP/M. It was originally developed for 8-bit computers and now works in a 16-bit environment under CP/M-86 or Concurrent CP/M. If you use CP/M-86, this may be your best choice for a Pascal compiler.

Recent versions of Pascal MT+ have been enhanced with the addition of a Speed Programming Package, which adds a full-screen editor and a syntax checking program. This means that many simple errors can be identified and corrected before running the compiler, saving you time and effort.

As in most versions of the language, Pascal MT+ includes many useful extensions to the original language definition. Byte, word, and string data types have been added, along with procedures to manipulate them. Dynamic array handling and the ability to include in-line assembler code has also been added. A modular compilation system is useful for assembling large programs with separately compiled modules.

The documentation does not attempt to teach you Pascal, but assumes that you will use other refer-

ence and tutorial books to learn the language. It primarily covers differences between Pascal MT+ and other versions of Pascal, as well as operational instructions on how to use Pascal MT+. This makes the documentation more difficult to use. A Pascal MT+ for the Model 100 should be available soon.

Pass Make information from one process or program available to another process or program. The sender may be a program or a person typing the data onto a command line to be "passed" to a program. A common example would be a BASIC program passing variables to a Machine language subroutine using CALL.

Password Used for identification and security purposes on a computer system. Each user is assigned a specific set of alphanumeric characters to gain entrance to all or part of the computer system.

PASTE Command Key. PASTE inserts the current contents of the paste buffer to the current cursor position in an open TEXT or BASIC file. To get text or program lines into the paste buffer enter TEXT mode and Define (F7) and Cut (F6) or Copy (F5) the text block you want in the buffer. Once there, the contents of the paste buffer remain unchanged until you alter them by putting new text into the buffer. You can copy the same text numerous times using the PASTE key. Turning off the computer does not affect paste buffer contents as long as the AA or nicad batteries continue to supply power for maintaining the RAM memory contents. See TEXT Mode.

Patch To debug or alter a software program, a section of code is inserted into the program which changes the control functions of the program.

PAUSE To temporarily pause execution of a BASIC program, press the BREAK/PAUSE key once. Pressing BREAK/PAUSE twice resumes execution of the paused program from the place execution was interrupted.

Pause To freeze the screen briefly during BASIC program execution while the operator using your program reads a message, write a delay loop after you print the message:

```
1000 FOR Y = 1 TO 2000
1010 NEXT Y
```

To freeze the screen, enter a dummy input statement and instruct the operator to press ENTER. The input variable need not be used in your program:

```
1000 INPUT "Press ENTER to continue";A$
```

PC Program Counter; or Printed Circuit; or Personal Computer.

PCHL 80C85 Assembly Language Instruction. Jump H and L indirect. Move H and L to PC. The contents of register H is moved to the high-order eight bits of register PC. The contents of register L is moved to the low-order eight bits of register PC. The addressing mode is register. No flags are set.

PCS Personal Computing System.

PE Parity Error. If the parity status bit goes to 1 in odd parity or 0 in even parity, an error is produced, a flag set, and a message displayed.

PEEK BASIC Function. Returns the byte value read from the specified memory position. The format is:

`<variable> = PEEK(<numvar>)`

`<numvar>` is an integer in the range 0 to 65024 which indicates the address of the memory location to be read.

The value returned is in the range 0 to 255. See Poke.

PerfectData Micro Maintenance Kit* A general purpose microcomputer care kit providing the cleaning supplies necessary for the small business computer or personal computer. It is intended for periodic preventative maintenance and cleaning of flexible disk drives, tape drives, CRT screens, printers, and keyboards. Included is a flexible disk drive head cleaning diskette and special cleaning solution, CRT screen cleaner, cleaning wands, lint-free cloths, a brush, anti-static spray to protect sensitive circuits from damage caused by static charge. An instruction book explains each component and assists you in planning an effective preventative maintenance program. Innovative Computer Products.

PerfectData Type Element Cleaning Kit* A simple, practical, and low-cost way to clean print-wheel type elements found on the majority of information-processing printers. Innovative Computer Products.

PerfectData Video Display Cleaning Kit* Cleans terminal screens and keyboards. Included are two 4-ounce bottles of video display cleaning solution, pump spray dispenser, and fifty cleaning cloths. The solution contains a static neutralizing agent which prevents build-up of electro-static charges that attract dust and dirt. The cleaning cloths included in the kit are specially designed to eliminate foreign particles and contaminants without scratching plastic surfaces. Innovative Computer Products.

Period . Filename/Extension Delimiter. In a filename, the colon (:) and period (.) are used to

delimit device name, and extension respectively. For example, "RAM:TEST.DO" specifies a RAM file named TEST stored in ASCII (document) format, as indicated by the .DO file extension.

Peripheral Any external device connected to a computer and controlled by it.

Personal Computer Generally, a single-user computer system which is inexpensive and can fit on a desktop.

PG1000 Color Printer* A color printing system using silent ink-jet technology. Precision images containing over 125 shades of color can be produced on standard computer paper. The specially formulated ink is contained in a snap-in, disposable cartridge. The PG1000 package includes printer interface hardware, cables, software drivers, 500 sheets of paper, and ink cartridge. Printacolor, Inc.

Phase Measured in degrees, a phase is the difference between the 0 crossing point of a reference waveform and that of the measured waveform.

Phase Locked Oscillator PLO. A circuit which is phase locked to recover data in a diskette drive controller. The PLO steadies the separate data and clock bits.

Phone Number, FIND a In TEXT mode, create an address file called ADRS.DO that Telcom can use to locate and retrieve phone numbers for manual or automatic dialing. The ADRS.DO file entries should be entered with the name first, then a phone number enclosed in colons. To enter Telcom, place the cursor over Telcom on the Main Menu and press ENTER. This displays the current telecommunications parameters on the top line of the screen and the prompt "telcom:" on the second line. To retrieve a phone number, press F1, the Find key. Enter the name of the person you want to call when the words "Telcom:prompt" appear. The program returns to the ADRS.DO file to look up that name. If it is not found, the prompt "telcom" is redisplayed. If the string is found, the program displays it on the screen. Any part of the address entry following the second colon in the phone number sequence won't be displayed. The temporary function key definitions Call, More, and Quit are displayed on the eighth screen line, above F2, F3, and F4, respectively. Particularly useful if you have two different numbers for one person, each in separate entries, is F3; it searches for the next occurrence of the person's name in the ADRS.DO file. F4, the Quit key, finds a different name and number. You are ready to autodial with the modem

PIC • Port

once you have the number you want to call displayed on the screen.

PIC Priority Interrupt Controller. A chip that manages interrupts and gives vectored interrupt capability to an MPU which does not have this feature built in.

Pin-Compatible Computer systems whose circuits have leads or pins with identical functions.

Pipelining When a computer gets the next program instruction before finishing the last instruction, thereby increasing the speed of the processor.

Pixel The smallest addressable unit in a video graphics display. The Model 100 contains two hundred and forty pixels across and sixty-four pixels down. Coordinate pixel addresses used in the PSET, PRESET and LINE statements begin at zero and end at two hundred and thirty-nine and sixty-three for the X and Y coordinates, respectively.

Pixel Graphics BASIC. PSET is used to turn on the pixel at a specified location on the LCD screen. The format is:

PSET(<xcoord>,<ycoord>)

<xcoord> and <ycoord> are the x and y coordinates of the pixel to be turned on. Values for the x coordinate range between 0 and 239. Values for the y coordinate range between 0 and 63. If you entered:

PSET(0,0)

the upper leftmost pixel would be turned on.

PRESET turns off the pixel at a specified location on the LCD screen. The format is:

PRESET (<xcoord>,<ycoord>)

<xcoord> and <ycoord> are the x and y coordinates of the pixel to be turned off. Values for the x coordinate may range between 0 and 239. Values for the y coordinate may range between 0 and 63. If you entered:

PRESET (0,0)

the upper leftmost pixel would be turned off. See PSET.

Plotter A computer-controlled mechanical device which draws images on a screen or printer.

Plug-Compatible Devices or components which use the same plugs and therefore can be used interchangeably without modification within the computer system.

PMD-100 Portable Micro Drive* Mass storage that operates nearly as fast as a disk system. Complete and ready to plug in, this wafer tape storage unit is battery operated with a rechargeable battery pack.

It is lightweight and fully portable. The operating system is ROM based and the unit has an optional RAM buffer. Holmes Engineering, Inc.

Pointer A data item whose contents is the address of another data item.

POKE BASIC Statement. Places a byte into a memory location. The format is:

POKE <memadd>,<byteval>

<memadd> must be in the range 0 to 65535. It indicates the address of the memory location where the data is to be written.

<byteval> is a decimal numeric expression representing the data to be written to the specified location. It must be in the range 0 to 255.

The complementary function to POKE is PEEK. POKE and PEEK are useful for efficiently storing data, loading Machine language programs, and passing arguments and results to and from Machine language subroutines. BASIC does no address checking, so use caution when assigning address locations. Never POKE data in BASIC's stack, BASIC's variable area, or your BASIC program.

Polling The status of each device is inquired according to a schedule.

POP A programming instruction (often Assembly language) where the last word is removed from the top of a stack.

POP PSW 80C85 Assembly Language Instruction. POP Processor Status Word. The contents of the memory location, whose address is specified by the contents of register SP, are used to restore the condition flags. The contents of the memory location, whose address is one more than the contents of register SP, are moved to register A. The contents of register SP are incremented by 2. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

POP <rp> 80C85 Assembly Language Instruction. The contents of the memory location, whose address is specified by the contents of register pair SP, are moved to the low-order register of register pair <rp>. The contents of the memory location, whose address is one more than the content of register SP, are moved to the high-order register of register pair <rp>. The contents of register SP are incremented by 2. Register pair <rp> = SP may not be specified. The addressing mode is register indirect. No flags are set.

Port A physical I/O connection. A port is an address providing a connection between the computer's internal processor and an external device.

Ports are used to attach input and output devices. Input/output ports in the Model 100 include an RS-232C serial port, a centronics standard parallel printer port, a Hewlett-Packard HEDS-3000 compatible bar code reader port, two female audio din plug interfaces for modem and cassette connectors, and a forty-pin external bus port.

PortaCalc* An electronic spreadsheet designed for small business owners. Graphically displays calculations such as addition, subtraction, division, and multiplication, and more extensive computations, such as percentage, exponentiation, absolute values, and averages. Recalculation of the entire file can be done with a single keystroke. The fourteen columns and twenty-six rows give versatility for user-defined fields, each field consisting of nine digits.

There are seven function keys for easy manipulation of data. These save, load, and print files, automatically return home, and display view or menu upon command. An extensive instruction manual is included for additional help. 24K; cassette. Skyline Marketing Corp.

PortaDex* Transforms labels and values into DIF format, which is used for the universal interchange of data. It is run simultaneously with PortaCalc* and Telcom. The user is prompted for each consecutive step and must supply information such as baud rate, word length, and file name. 24K; cassette. Skyline Marketing Corp.

PortaFin* Calculates present and future value, interest, and annuities for loans and investments. Current cash flow data is readily available when this financial program is combined with PortaCalc*. Keyboard, memory, or cassette can be used for input. 24K; cassette. Skyline Marketing Corp.

PortaFolio* Keeps calculated records of stock and bond values, yield to maturity, Macaulay's duration, and more. A detailed instruction manual included. 24K; cassette. Skyline Marketing Corp.

PortaMax* A powerful and easy-to-use linear program. Locates the most favorable blend of grouped constraints. Includes a detailed instruction manual. 24K; cassette. Skyline Marketing Corp.

PortaMed* Functions as an electronic clipboard for updating and generating medical reports. Members of the medical profession are able to use their time more efficiently by organizing important records for easy retrieval. 24K; cassette. Skyline Marketing Corp.

PortaPrint* A printing utility that works as a word processor in formatting and printing text files. You

create a text file and save it for later retrieval. Three character codes are available which enable page, line, and margin control. After accessing the print mode, you choose your format specification options. Each has default values if unanswered. Printing begins automatically once the file name is entered. 24K; cassette, printer. Skyline Marketing Corp.

PortaStat* A statistics program designed to work with PortaCalc*. Regression analysis, correlation, and covariance abilities aid the user in many business applications. Keyboard, cassette, or RAM can be used to input data. 24K; cassette. Skyline Marketing Corp.

POS BASIC Function. Returns the current cursor column position on the LCD. The format is:

<variable> = POS(<dummysnum>)

<dummysnum> is a dummy argument.

The current horizontal (column) position of the cursor is returned. The returned value will be in the range 0 to 39 as the LCD is forty columns across. CSRLIN can be used to find the vertical (row) position of the cursor. See CSRLIN and LPOS.

Position of Carriage BASIC. See LPOS.

Position the Cursor The cursor is the underline symbol (or with some programs a different symbol such as a square) which appears on the screen to let you know where an action (such as entering, deleting, or inserting a character) takes place. The cursor moves to the right as you type and backs up to the left when you press backspace.

The Model 100 has four half-size cursor movement keys located on the right end of a row of special keys, directly above the top row of the standard keyboard. The keys are labeled, left to right, ←, →, ↑, and ↓. Pressing any one of these cursor keys once moves the cursor one character space in the direction of the key label. As with the other keyboard keys, the cursor keys automatically repeat if held down. The cursor keys move the cursor in basically the same way in all four applications programs, BASIC, and the Main Menu. In BASIC Entry Mode, SCHEDL and ADDRESS programs, and Telcom Entry Mode, you won't be able to move the cursor in all directions. For instance, in BASIC Entry Mode, the ↑ and ↓ cursor keys don't work because cursor movement is limited to the current line prior to its being entered to BASIC. In this instance, backspacing erases the character the cursor backspaces over. In each case, once you enter the current line, it cannot be changed. In BASIC EDIT, TEXT, Main Menu, and Telcom the cursor keys can be used to move the cursor in all four directions without erasing any characters.

Positive Logic • PRESET

In TEXT and BASIC EDIT, the cursor key functions can be expanded by pressing a cursor key along with either the SHIFT or CTRL key. SHIFT and the ↑ or ↓ key moves the cursor to the top or bottom, respectively, of the screen. If the cursor is already there, the next screenful in the given direction scrolls onto the LCD. CTRL plus ↑ or ↓ takes the cursor to the top or bottom of the file, displaying the first or last eight lines, respectively, on the LCD. SHIFT and → moves the cursor to the first character of the next word to the right on the same line. SHIFT and ← moves the cursor to the first character of the next word to the left. CTRL and → or ← moves the cursor to the right or left end of the current line. These key combinations are only valid in TEXT and BASIC EDIT.

Positive Logic Voltage level most positive in the computer system is the true level; the false level is the voltage level closest to zero.

Postfix Notation system where the operator follows the manipulation symbols used in the program.

POWER BASIC. Turns off the computer during BASIC program execution or sets the period of user inactivity, after which an automatic power down occurs. The original default value is ten minutes before a power down occurs. The format for changing the default time is:

POWER<numex>

<numex> is numeric expression with a value ranging between 10 and 255. Units in this case are tenths of minutes. POWER 10 sets the automatic power down to one minute. POWER 100 sets it at 10 minutes. Once changed, the power down time remains as specified until you change it or erase all RAM memory.

To disable the power down option (not recommended unless plugged in using an AC adapter) specify:

POWER CONT

POWER can also be executed as part of a BASIC program. You can terminate program execution and power down using two formats, with two results:

POWER OFF

turns off the computer and returns the user to the Main Menu the next time the computer is powered on.

POWER OFF, RESUME

turns off the computer and returns program execution to the line following the POWER OFF, RESUME the next time the computer is turned on.

Power Down To preserve the state of the processor and prevent damage to it or to connected per-

ipherals when the power fails or is shut off, a computer takes certain steps. The Model 100 has an automatic power down feature that executes after ten minutes of computer inactivity. You may change the duration before an automatic power down disables the feature, or automatically power down from within a BASIC program using the POWER command.

Power On Turn the power switch to the ON position or otherwise supply electric current to a device.

Power ON/OFF Switch Located on the right side of the computer, near the front. If the computer receives no input for ten minutes and is not printing a file or running a program, it turns itself off to conserve power. This is particularly useful if you use batteries. You can change the interval of inactivity tolerated before an automatic power down to between one and twenty-five minutes using the BASIC POWER command, or disable the feature entirely. To turn the computer back on after it turns itself off, turn the computer first off and then on again. It is a good practice to turn on the computer before turning on any attached peripheral devices and turn all peripherals off before turning the computer off.

Power Supply The unit that converts the voltage from the electrical supply to the voltages which the computer elements use. See AC/DC Adapter Unit, Batteries.

Power Up When the power is turned on or restored after a power failure, the processor and peripherals are initialized so that program execution can start.

Power-Fail Restart A device which detects a drop in the voltage and signals the processor. The processor still has several milliseconds to preserve the registers in a battery backup memory, allowing automatic resumption of processing when the power is restored.

Power-Up Diagnostic The self-test a computer performs each time the power is turned on.

Pre-Processor A program or mechanical device which prepares data for further processing.

PRESET BASIC Statement. Turns off the pixel at a specified location on the LCD screen. The format is:

PRESET (<xcoord>, <ycoord>)

<xcoord> and <ycoord> are the x and y coordinates of the pixel to be turned off. Values for the x coordinate may range between 0 and 239. Values

PRINT • PRINT Command Key

for the y coordinate may range between 0 and 63. If you entered:

PRESET (0,0)

the upper leftmost pixel would be turned off.

PRINT BASIC Statement. Displays data on the LCD screen. The format is:

PRINT <list of expressions>

or

?<list of expressions>

<list of expressions> is a list of numeric and/or string expressions, separated by commas, or semicolons. You must enclose any string constants in the list in quotation marks.

A PRINT statement alone displays a blank line. The question mark (?) is a shorthand way of entering PRINT.

The punctuation in the <list of expressions> determines the spacing of the printed items. BASIC divides the screen line into zones of fourteen spaces each, starting in the zero column. In the <list of expressions>, a comma (,) causes the next item to be printed starting in the first space of the next print zone. A semicolon (;) causes the next value to be printed immediately after the last value.

If the value to be printed is longer than forty characters, BASIC prints as much as possible on the current line and continues to print the remaining values on the next line. BASIC automatically returns the cursor to the beginning of the next line after printing all the values on the <list of expressions>. Printed numbers are followed by a space. Positive numbers have a space in front of them and negative numbers have a minus sign in front of them. Trailing zeros after a decimal point are not printed. No extra blanks are printed with string variables.

LPRINT statement uses the same format techniques but displays the values on an attached printer. See LPRINT and LPRINT USING.

Print, Compressed To set the EPSON RX-80 printer for this small type size, enter the BASIC statement:

LPRINT CHR\$(27); CHR\$(15);

either in a program with line numbers at the point you want to start printing in the small type, or directly, without a line number. This gives 136 characters on the 8 inch line, or about 17 characters-per-inch. To turn it off, enter:

LPRINT CHR\$(27); CHR\$(18);

See Type Formats.

PRINT, Move Cursor Down On Screen To move the cursor down one space, enter BASIC statement:

PRINT CHR\$(10)

Press the cursor down (↓) key. This only moves the cursor; it does not erase data on the screen.

PRINT, Move Cursor Left On Screen To move the cursor left one space, enter BASIC statement:

PRINT CHR\$(8)

From the keyboard, DEL/BACKSPC or the cursor left key (←) moves the cursor one space left. The DEL/BKSP key always deletes the character over which the cursor backspaced. If you use the cursor left key to backspace, the characters backspaced over are deleted in all modes except TEXT and BASIC EDIT. In these modes, backspacing with the cursor left key (←) is non-destructive.

PRINT, Move Cursor Right On Screen To move the cursor right one space on the screen, enter BASIC statement:

PRINT CHR\$(28)

From the keyboard, enter any character, press the spacebar, or press the cursor right (→). This moves the cursor one space right for each character entered; it does not erase the screen.

PRINT, Move Cursor Up On Screen To move the cursor up one space on the screen, enter BASIC statement:

PRINT CHR\$(30)

From the keyboard, press the cursor up key (↑). This moves the cursor one space up; it does not erase the screen.

Print, Normal To return to normal ten character-per-inch print size, you must turn off all non-standard print options. See Type Formats.

Print BASIC Program Line on Printer See LLIST.

Print Blank Line on Printer LPRINT prints a blank line (that is, feeds the paper up one line and returns to left margin) so you can space your printout format neatly.

PRINT Command Key The PRINT key has no effect unless your computer is attached to a printer with a standard Centronics style interface using the Model 100 Printer Cable (RS 26-1409). Pressing PRINT causes the current LCD screen contents to be dumped to the printer and, if the function key label display is on, it is printed as well. Pressing the SHIFT and PRINT keys together prints the entire contents of the currently open BASIC or text file. In TEXT, pressing this key combination first produces the prompt "width:" followed by the currently defined print width. The initial default value is forty characters-per-line, but you can change that value. Valid line width values range between 10 and 132. If

Print Current Screen • PRINT USING

you are satisfied with the value given by the width prompt, press ENTER to begin printing the file at that width. To interrupt printing, press the SHIFT and BREAK/PAUSE keys at the same time. See Print Current Screen.

Print Current Screen The PRINT command key prints the current contents of the LCD screen on an attached printer. This is valid in any program and all applications at any time. See Control Keys.

Print from BASIC See LPRINT, LLIST, and SAVE ILPF.

Print Lines, Spacing BASIC. For one or more spaces between fields printed by your BASIC programs, use a literal of spaces like: " ". To get several spaces between the printed values of A\$ and B\$, use BASIC statement:

```
LPRINT A$;" ";B$
```

See also Print Zones.

PRINT USING Formats and displays the values in the expression list on the LCD, using the formats specified in its <format string>. The format is:

```
PRINT USING "<format string>";  
          <list of expressions>
```

<list of expressions> is the list of the expressions whose values are to be printed. They can be numeric or string variables or constants. They are separated by commas (,) or semicolons (;). The punctuation of the expression list determines the spacing of the printed items. LCD screen lines are separated into fourteen spaces. A comma in the expression list causes the next item to be printed starting in the first space of the next print zone. A semicolon (;) causes the next item value to be printed immediately following the previous item.

"!" tells the program to expect string data and instructs it to display only the first character in the string. For example:

```
PRINT USING "I"; "BOB"
```

prints B.

"nnn," where each n represents a space between the quote marks, tells the program to expect string data and to display on the LCD two characters from the string, if there are no spaces between the quote marks. If there are spaces between the quotes, then two string characters are displayed plus as many more characters as there are blanks. For example:

```
PRINT USING " "; "Marcia"
```

displays Marc because there are two blanks between the quote marks and BASIC adds two.

"####" tells the program to expect numeric data and to display on the LCD as many digits as there are hashmarks (#). If there are fewer digits than marks,

the digits are padded with zeros to their left until they fill the field, right-justifying the digits. If there are more digits than there are marks, BASIC displays all the digits in the expression list, preceded by a percent sign (%) to show that the number of digits has overflowed the field specifier.

A decimal point can be located at any position in the field. If the formatting specifies a number of digits to precede the decimal point, the digit is always represented, with blanks if necessary. Numbers to the right of the decimal are rounded, if necessary, to display them in the format specified. For example:

```
PRINT USING "####";34
```

displays the number given in the expression list (34) to four spaces, padding the left with blanks. The result is two blanks, then 34.

```
PRINT USING "##";3456
```

```
PRINT %3456
```

"+" tells BASIC to expect numeric data, and can be appended to other field specifiers at the beginning or end. It causes the value of the expression to be preceded or followed by its algebraic sign (+ or -). If the value of the number given is positive, then a plus sign is displayed on the same side of the expression value as it is in the field specifier. If the expression is negative, then a minus sign (-) is used in the same way. For example:

```
PRINT USING "+####";-45
```

displays the number in the expression list (-45) to four digits and prefaces it with its algebraic sign, in this case a minus sign (-). The algebraic sign is not considered a digit. The result is two spaces then -45.

"-" tells BASIC to expect numeric data and can be appended to other field specifiers at the beginning or end. If the algebraic value of the expression formatted is negative, BASIC places a negative sign (-) on the same side of the number that the minus is on in the field specifier. If the expression has a positive value, and the minus sign follows the rest of the format specifier, BASIC displays a blank at the end, rather than a minus sign. If the value of the expression is positive, and the minus sign precedes the rest of the field specifier, BASIC prints a minus sign first and then the number value of the expression as formatted by the rest of the field specifier. For example:

```
PRINT USING "###-";78
```

displays (78-) one space and the number 78; directly followed by a minus sign.

```
PRINT USING "###-";45
```

Prints (45) the number 45 right-justified by a blank in three spaces, followed by a blank, caused by the

minus sign in the field specifier. However:

PRINT USING "***###";26

displays (**26), the number shown three spaces and right-justified using three asterisks. The first two asterisks shown on the left are produced by the two asterisks in the format string. The third is a space that has become an asterisk. The space which would have been produced by padding two digits into three spaces (###), is transformed into an asterisk by the presence of the first two asterisks in the format string.

"\$\$" tells BASIC to expect numeric data and inserts a dollar sign (\$) to the immediate left of the number as shown within its specified format. Only one of the dollar signs is displayed. The second one is inserted as a blank to the left of the dollar sign. The double dollar (\$ \$) specifier may be used in conjunction with other field specifiers. For example:

PRINT USING "\$ \$###";56

displays (\$56), two blanks, one from an undisplayed dollar sign, and one as padding from the third hashmark (#). Follow the displayed dollar sign and the right-justified two digit numeric value.

"**\$" tells BASIC to expect numeric data and works like a combination of the "***" and the "\$ \$" field specifiers. The two leading asterisks in the field specifier are printed and also turn any leading blanks in the formatted number to asterisks. The dollar sign (\$) in the field specifier is inserted to the immediate left of the numeric value. "**\$" work in conjunction with other field specifiers. For example:

PRINT USING "**\$###";45

displays (**\$45), three asterisks followed by a dollar sign, followed by the numeric value. Two of the three asterisks printed are from the two asterisks in the field specifier. The third is a space turned asterisk. The space which would have been produced, by padding two digits into three spaces (###), is transformed into an asterisk by the presence of the first two asterisks. The dollar sign as given by the field specifier is inserted to the immediate left of the numeric value.

Placing a comma to the left of a decimal point among the field specifiers causes a comma to be displayed, separating every third digit left of the decimal point. A comma also specifies another digit position. If a digit to the immediate left of a potential comma is blank, the comma is shown as a blank, too. When used, the comma must be sandwiched in the format string between other field specifiers, such as \$ \$**, and #.

"AAAA" four carets are placed after the digit position characters to indicate exponential format. E+/-

nn or D+/-nn is printed in the four spaces allowed by the four carets. You may specify any decimal point position. Significant digits are left-justified, and the exponent is adjusted accordingly. One digit position left of the decimal point is used to print a space or a minus sign if a leading or trailing + or - has not been designated.

Print Zones Each group of 14 spaces across the print line is called a print zone. A comma (,) in an LPRINT list of items to be printed means "start printing the following item at the start of the next print zone." The semicolon (;) means the next item to print is immediately after this one, without a single space between. The print zones begin in columns 0, 14, 28, 42, 56, and 70. If you use semicolons, BASIC brackets numeric data with a leading and following blank if the number is positive, and a leading unary minus and a trailing blank if the number is negative. BASIC adds no spaces to string values and does not print zeros to the right of the decimal. See Print Line, Spacing.

PRINT# and PRINT# USING BASIC Statements. PRINT# writes data sequentially to a file. PRINT# USING writes to a file, but USING allows control of the format of the file. The format is:

PRINT#<filenum>,[USING <v\$>;]
<list of expressions>

<filenum> is the number used when the file was opened in an OPEN statement.

<v\$> is a string expression made up of formatting characters.

<list of expressions> is a list of the numeric and/or string expressions that will be copied to the file opened as <filenum>.

PRINT# writes data to the file as it would appear on the screen with no compression. For this reason, your data should be edited carefully for file input. Numeric expressions should be delimited by semicolons. If commas are used as delimiters, the extra blanks that are inserted between print fields are also written to the file. String expressions must be bracketed by double quotes. To format the string expressions correctly on the file, it is best to use explicit delimiters in the <list of expressions>. See PRINT USING.

PRINT* A word processor providing page, margin, and centering controls which are easily specified for individual user specifications. Cassette. Micro Computer Services.

Printexlex Printer* A printer designed especially for use with the smaller "briefcase size" portable computers. This compact, portable, thermal print-

Printer—Top of Page • Processor-Bound

er operates at 160 characters-per-second. It features two interfaces: the standard Centronics parallel and the RS-232C standard serial. The serial interface baud rates are switch selectable between 1100 and 4800. Line width is forty columns, and graphic capabilities are featured. Atlantic Northeast Marketing, Inc.

Printer—Top of Page To form feed or advance to top of the next page on the EPSON RX-80, enter the BASIC statement:

```
LPRINT CHR$(12)
```

Or use the "top of form" or "form feed" manual control button (FF) on the printer. You may then need to adjust the paper in the printer so it actually is at the top of a page as defined by the perforations. In a program, you may want to provide instructions to the operator and a pause to allow for adjustment of the paper. See Pause.

Printer—6 Lines-Per-Inch See Lines-Per-Inch (6) and Type Formats.

Printer—8 Lines-Per-Inch See Lines-Per-Inch (8) and Type Formats.

Printer—7/72 Lines-Per-Inch See Lines-Per-Inch (7/72) and Type Formats.

Printer, Advancing One Line On To advance one line on the printer (space up) without a carriage return, enter BASIC statement:

```
LPRINT CHR$(10)
```

or use the "line feed" button (LF) on your printer. Entering just LPRINT gives a line feed—both a space up one line (line feed) and a return to left margin (carriage return).

Printer, How To Hook Up The Model 100 can be connected to any Centronics style parallel printer using the Model 100 Printer Cable (RS-26-1409). The larger end of the printer cable connects to the parallel port of the printer. The smaller end of the printer cable inserts into the printer connector port on the rear panel of the Model 100, between the RS-232C and phone connector ports. The small end of the cable that attaches to the computer has one side marked UP. It is a good idea when using peripherals such as the printer to turn on the computer first, then turn on the peripheral, such as the printer. When you turn off the system, turn off the peripherals before the computer.

Printer, Line Length on See PRINT Command Key.

Printer Carriage Position BASIC. See LPOS.

Printer Control Codes BASIC. To set the Epson RX-80 printer's print size, strike method, or number of lines-per-inch, the non-standard type format you want must be turned on by sending control codes to the printer. This can be done in BASIC by entering a BASIC CPRINT line containing ASCII control codes, and other special codes. See Type Formats.

Printer I/O BASIC Commands

I/O Commands	Function Performed
LCOPY	Prints current screen contents on the printer
LLIST	Prints all or a part of the current contents of BASIC memory on the printer
LPRINT	Prints data on the printer.
LPRINT USING	Prints data on the printer using a specified format
LPOS	Returns the current position of the printer head
TAB	Prints the next data beginning at the line position specified

Printer Page Length, Setting To set page length on the EPSON RX-80 to <n> lines per page, enter BASIC statement:

```
LPRINT CHR$(27);"C";CHR$(<n>)
```

<n> may range between 1 and 127 and represents lines-per-page.

Printer Type Formats BASIC. To set print size, strike method, or number of lines-per-inch, the non-standard type format you want must be turned on by sending control codes to the printer. See Type Formats.

Probe An electrical device which, when it contacts a circuit point, allows a test meter to test the circuit's connection and power.

Procedure A part of a program which helps the program's structure, readability, or reliability. The procedure is a separate function of the program and could be incorporated into a subroutine.

Processor-Bound When the speed of the processor limits the processing speed of the program, the computer is processor-bound.

Program A sequence of instructions specifying a process for manipulating data. Programs can be written in many languages of different "levels." The level reflects how much additional work the processor must do before the program can be executed. BASIC is a high-level language which requires that all programs be run through a compiler or interpreter. Assembly language is mid-range, closer to the binary code of the computer, but still in need of processing through an Assembler before execution. The lowest level are programs written in binary or hexadecimal code which are directly executable by the microprocessor. Before Assembly language programs were developed, all programs were written in binary code—a tedious, time-consuming task.

Program—Version or Release Programs and software packages are changed from time to time to correct errors or add new capabilities. To keep programs from being in a constant state of flux and to simplify distributing modified programs to users, a number of changes are made, tested, and packaged as a new version or release of the program. Versions or releases are typically numbered to help programmers support a program by letting them know with which version the user has encountered problems. Numbering also helps users know which changes are effective in the version they possess. A typical numbering system is: 1.0 for the first version, 1.1 for the first minor revision, 1.2 for the second minor revision, etc. When a major change or large number of minor changes have been made, a new number may be assigned, as Version 2.0.

Program, BASIC See RUN, and LOAD.

Program, Object The Machine language instructions which result from translation of a source program by a compiler or Assembler. Object programs for the Model 100 have a .CO (Command) file extension.

Program, SAVE BASIC. To save a BASIC program to a specified device. The formats are:

SAVE "[<dev:>][<filename>]".A]

or

SAVE "<dev:> [<r>] <w> <p> <s>

or

SAVE "<dev:>

The format you use depends upon the device to which you are SAVEing.

<dev> in the first format may be RAM: or CAS:, which save the BASIC file to either RAM memory or cassette tape; MDM: or COM:, which transmit

the BASIC file, via the built-in modem or the RS-232C interface, to another computer; LCD: or LPF:, which display the data on the screen or print it to the printer. If you omit the <dev> specification, BASIC assumes you are saving the BASIC memory contents to a RAM file and requires that you specify a <filename>.

<filename> is the six character name, beginning with a letter, that is assigned to the BASIC file you are saving to RAM or cassette. If <dev> is RAM: or CAS: you must specify a <filename>. No file extension is necessary.

If you specify the device as MDM: or COM:, the file is transmitted to another computer, in serial format, through the built-in modem (MDM:) or RS-232C interface (COM:). In this case, BASIC requires that you specify the communications parameters to be used (format 2). They should match the communications parameters that the computer on the receiving end is using. Each parameter is defined by a single-character value which must be entered in the sequence <t> <w> <p> <s>. Each parameter has its own valid range of values. A value from the appropriate range should be entered to BASIC for each parameter. Parameter values for each code are:

<r> is the baud rate or bits transmitted per second. Valid values for this parameter are:

- 1 = 75 baud
- 2 = 110 baud
- 3 = 300 baud
- 4 = 600 baud
- 5 = 1200 baud
- 6 = 2400 baud
- 7 = 4800 baud
- 8 = 9600 baud
- 9 = 19200 baud

If the communications file is to be transmitted using the modem (MDM:), then the <r> communications parameter should be omitted. The modem transmits at 300 baud by default. If this default value is altered, even if the alteration is to 300 baud, then the modem is disabled.

<w> represents word length in bits and may be set to values 6, 7, or 8. <p> represents parity and may be set to:

- E—meaning even
- O—meaning odd
- I—meaning ignore
- N—meaning no parity

 Stop bits. Valid values are 1 for one stop bit or 2 for two stop bits.

<s> Enables and disables XON/XOFF status. E enables and D disables.

If you specify the device as LPT: or LCD:, the current contents of BASIC memory are sent to the printer or the LCD screen. In this case, it is sufficient to specify the device only. No <filename> or communications parameters are necessary.

SAVE "CAS:<filename>" is the same as CSAVE "<filename>"

Program, Source A file which contains data to be processed by a language processor or interpreter. For example, if you write a BASIC program called SAMPLE.BA, it is a source program. If you then submit it to a BASIC compiler to produce a fast object program, the BASIC compiler treats SAMPLE.BAS as its input data rather than as a program to be executed. The BASIC compiler produces an object file with extension .CO which is in Machine language.

A BASIC source program looks more like a document text file (.DO) than like an executable program (.CO). So it is how a file is used (or usually used) that determines whether it is a data file, a program, or (in the case of a source program) sometimes one and sometimes the other. But in general, a program file contains some type of program instructions specifying how data is to be processed, while a data file contains the data that will be processed. See Programs.

Program Concatenation Adding one item on at the end of others to produce one longer data or program item. Also called data concatenation.

Program Counter A memory register which holds the address of the next instruction to be executed. The counter is incremented each time an instruction is executed. The counter can be modified through subroutines or calls so that the instruction next executed is different from that which directly follows in the program.

Program Diskette A program diskette is primarily used to store data processing programs. Data diskettes are used primarily or totally to store data rather than programs. To save space, a data diskette usually will not contain DOS programs. A system diskette is used to store DOS commands and related utility-type programs. Often, a diskette contains both data and the programs needed to process the data.

Program File Contains program instructions specifying how data is to be processed. If you write a BASIC program called SAMPLE.BA, this is a source program. If you then submit it to a BASIC compiler

to produce a fast object program, the BASIC compiler treats SAMPLE.BA as its input data rather than as a program to be executed. The BASIC compiler produces an object file, with the extension .CO, which contains a translation of your source program into the 80C85 Machine language.

A BASIC source program looks more like a document text file than like an executable program (.CO). It is how a file is used (or usually used) that determines whether it is a data file, a program, or in the case of a source program, sometimes one and sometimes the other. See Program; Program Source; Program, Object; and BASIC Compiler.

Program Integrity Ensuring that programs (or data) cannot be altered improperly. For example, in a payroll system, steps must be taken to ensure that employees cannot improperly alter their pay rates or hours worked. Data security consists of guaranteeing both data integrity and data secrecy or privacy.

Program Lines—BASIC, Deleting See EDIT.

Program Lines, LIST To display program lines on the screen, enter:

LIST

To display program lines from start up to line 100, enter:

LIST-100

To display program lines from line 100 to end, enter:

LIST 100-

To display program lines from line 100 to line 200, enter:

LIST 100-200

To display program line 100 only, enter:

LIST 100

To freeze the list while it is being displayed, press the PAUSE key once. To resume after pausing, press the PAUSE key a second time.

Program Operator See User, User-Friendly.

Programming Language A language that can be translated into Machine language and used to direct the computer to carry out functions.

PROM Programmable Read-Only Memory. A ROM which can be modified by the user. See ROM.

PROM Burner See PROM Programmer.

PROM Programmer An external device or module used to write user modified ROMs. The programmer may input data through a hex keyboard, binary paper tape, or directly through the microprocessing unit.

Prompt In general, a prompt is a visual signal from a program to "prompt" the operator to do something. The prompt should give some clue as to what the program requires (or allows) the operator to do in response to the prompt.

Prompts The prompts OK and ? are BASIC's ways of letting you know it is waiting for you to enter information.

OK is the prompt from BASIC indicating that you can now enter a BASIC command or statement.

? is the prompt from a program written in the BASIC language which tells you to answer a question. In this case, it is a good idea to include a descriptive prompt such as "Enter check amount" in the program so the operator knows exactly what should be entered in response. An example is:

```
100 INPUT "ENTER CHECK AMOUNT ",
    CHECKAMOUNT
```

This is displayed on the screen as:

```
ENTER CHECK AMOUNT ?
```

Propagate To go from one component in the computer system to another.

Propagation Delay The time the processor takes to pass a signal through one device on the system to another.

Proportional Spacing Where the printer allocates horizontal space according to a character's width, rather than having standard width space for all characters. Proportional spacing is more readable than fixed width type, and appears to be typeset.

Protect High Memory Location See CLEAR.

Protected Diskette See Write-Protected.

Protected Field Some data entry screens have areas that are reserved and cannot be modified by the user with keyboard entry.

Protocol The rules governing the exchange of information between computer systems.

PSET BASIC Statement. Turns on the pixel at a specified location on the LCD screen. The format is:

```
PSET(<xcoord>,<ycoord>)
```

<xcoord> and <ycoord> are the x and y coordinates of the pixel to be turned on. Values for the x coordinate may range between 0 and 239. Values for the y coordinate may range between 0 and 63. If you entered:

```
PSET(0,0)
```

the upper leftmost pixel would be turned on.

Pseudo-Instruction A user-defined directive which

is interpreted by the Assembler as one or more instructions.

Pseudo-Operation An operation code recognized by an Assembler but not corresponding to any Machine language instruction.

PSW Program Status Word. Contains the zero flag, carry flag, and other information relevant to the processor.

Pulse A square or gaussian-shaped voltage or current lasting a short period of time.

Pulser A circuit that, while under test, delivers a high current signal of short duration.

PUSH An Assembler instruction used to put a word on the top of a stack.

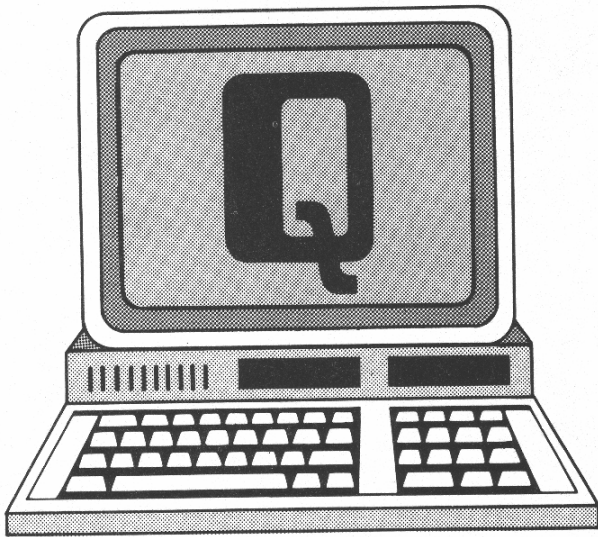
PUSH PSW 80C85 Assembly Language Instruction. PUSH Processor Status Word. The content of register A is moved to the memory location whose address is one less than register SP. The contents of the condition flags are assembled into a processor status word and the word is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. The addressing mode is register indirect. No flags are set.

PUSH <rp> 80C85 Assembly Language Instruction. The content of the high-order register of register pair <rp> is moved to the memory location whose address is one less than the content of register SP. The content of the low-order register of register pair <rp> is moved to the memory location whose address is two less than the content of the register SP. The content of register SP is decreased by two. Register pair <rp> = SP may not be specified. The addressing mode is register indirect. No flags are set.

Pushdown List Another name for a stack.

PUT* An address organizer designed for the business person. A maximum of sixteen categories are available for easy access of data. Two other files which keep detailed records of appointments and inventory can be obtained using this cassette.

User-specified screens are easily created for precise formatting. Special commands, such as find, sort, and search, may be executed in any category by specific function keys. 16K; cassette. Portable Computer Support Group.



two rows of pins on each side. Much smaller than a comparable DIP (Dual In-line Package).

Qwerty The traditional typewriter keyboard layout, named after the first six letters in the top letter row.

Q Codes. ASCII 81, HEX 51. q—ASCII 113, HEX 71.

Q Half-width of power spectrum of bandpass filter response in hertz, divided into the center frequency in hertz. Also, a register used as an accumulator extension, necessary for efficient multiply-divide operations. Q was not generally provided in earlier 8-bit CPUs. Newer 8-bit and 16-bit CPUs generally have a larger set of general purpose registers than the older 8-bit CPUs having an A register/Q register combination.

QA Quality Assurance.

Q-Bus The internal system bus of the DEC LSI-11 computer.

QC Quality Control.

QTAM Queued Teleprocessing Access Method (IBM term). IBM Mainframe communications method.

Quad Involving four entities, or a multiple of four.

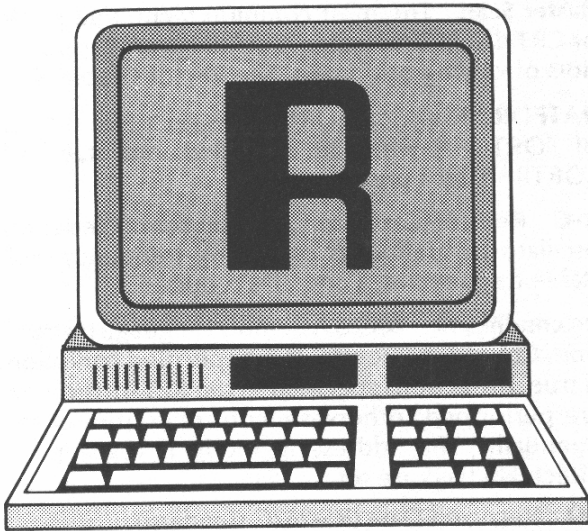
Quad-Density Specifies storage density of a disk medium. Quad-density stores four times as much information per disk as single-density.

Whereas double-density drives write forty-eight tracks-per-inch (TPI) of diskette surface (48 TPI), a quad drive writes the tracks twice as close together (96 TPI). This puts twice as much data within the travel distance of the drive arm.

Quadruple Density See Quad-Density.

Queue A data structure which contains data or tasks waiting to be processed.

QUIP QUad-In-line Package. An IC package with



R Codes. ASCII 82, HEX 52. r—ASCII 114, HEX 72.

R Reset. Also, Register, Request, and Ring Indicator.

Rack Mountable Describes equipment packaged for installation in a metal cabinet, or rack.

Radiation Hardening A quality assurance process used in the production of ICs to select circuits which are able to withstand radiation.

Ragged Margin TEXT, the built-in text processor in the Model 100, left-justifies the text it prints, but does not right-justify it, leaving a ragged right margin. See Justify.

RAL 80C85 Assembly Language Instruction. Rotate left through carry. The content of the accumulator is rotated left one position through the CY flag. The low order bit is set equal to the CY flag and the CY flag is set to the value shifted out of the high order bit. Only the CY flag is set.

RALU Register equipped ALU. A bit-slice ancestor (National Semiconductor).

RAM Random Access Memory. Memory is any device which can store information and allow it to be retrieved when needed. The Model 100 relies primarily on random access memory (RAM), read-only memory (ROM), diskettes, cassettes, and hard disks. Memory, by itself, is usually a reference to RAM. This is the general purpose, erasable and reusable memory located inside the Model 100. See External Memory, Memory, and ROM, Virtual Memory, RAM Disk, Spooling, Bank Switching, Memory Maps, Memory Addresses, Memory Segments, Bubble Memory, RAM Memory Cards, Disk, and Address.

RAM Cartridges See NEC PC-8201A—Memory Scheme.

RAM Disk A block of RAM memory used by a program to simulate a diskette drive. The program must modify or intercept all I/O to the “fictitious” drive and redirect it to the RAM memory. The RAM disk has its own drive letter and resembles a real diskette except that it is much faster.

RAM File, Create To create a new RAM file from TEXT, enter TEXT mode from Main Menu. When the prompt “File to Edit?” appears, specify a file name not already in RAM. This creates a new file using the name you specified. Pressing F8, the Menu key, automatically saves the new file to RAM with a .DO extension, which signifies that the file is stored in ASCII format.

To create a new RAM file for BASIC, use NEW to erase the current BASIC memory contents. Then write a program or load one from a device other than RAM (CAS: MDM: or COM:). Next, save the file to RAM. Press F3, the SAVE key, to enter in “SAVE” at which point you enter the name of the new RAM file. You can also enter the entire phrase manually from the keyboard. In either case, pressing ENTER saves the current contents of BASIC memory to a new RAM file, using the name you supplied and adding the file extension .BA to signify that it is stored in tokenized BASIC format. You can optionally choose to specify that the BASIC file be saved to RAM in ASCII format by including the ,A option of the save command. For example:

SAVE” text”,A

saves the file in BASIC memory to a RAM file named text, in ASCII format. In this case, BASIC adds the file extension .DO, to signify that the file is stored in ASCII format.

RAM File, Delete KILL deletes a file from RAM memory. The format is:

KILL “<filespec>”

<filespec> is the name of the RAM file as it appears on the Main Menu. It must include the two characters’ file extension. Although you must be in BASIC to use the KILL command, it deletes BASIC, TEXT, and Machine language RAM files. In fact, it is the only way to delete the file without erasing all of RAM memory.

Occasionally, when fewer than 200 bytes of free memory are left, BASIC will not delete the file specified by the KILL command. You must open the file and delete lines or blocks until there is enough free memory to kill the remainder of the file. You can also temporarily store another file in the paste

RAM File, Rename • READ

buffer, kill the target file, and paste the file in the paste buffer back into the RAM file it came from.

RAM File, Rename BASIC. RAM file names are listed on the Main Menu or by entering the BASIC FILES command while in BASIC mode. You can rename any RAM file on the listing by using the BASIC NAME...AS command. For example, if you had a RAM file named ORION.DO and you wanted to change its name to HOMER.DO you would do that by entering BASIC mode and entering the command line:

```
NAME "ORION.DO" AS "HOMER.DO"
```

Both the old and new file names must be enclosed in double quote marks and the file extension must be included and cannot be changed. The second (new) file name you give must not be the name of an existing RAM file.

RAM File, Rename NAME changes the name of a RAM file. The format is:

```
NAME "<filename1>" AS "<filename2>"
```

<filename1> is the name of the RAM file that you want to change.

<filename2> is the new name of the RAM file.

The file declared by <filename1> must be in RAM and the file declared by <filename2> cannot be in RAM. The two letter extension of the RAM files must be used and cannot be changed in renaming the files. Filenames can also optionally be specified as "RAM: <filename>" for greater clarity.

RAM Memory Card A card containing RAM memory to expand the computer's main memory.

RAM Pack A package containing expansion RAM to mount external to the computer. If mounted inside the computer housing, it would be called a RAM card. See RAM Memory Card.

Random Access An access method where each data item can be retrieved directly by an address computed from the data.

Random Number BASIC. See RND.

RAR 80C85 Assembly Language Instruction. Rotate right through carry. The content of the accumulator is rotated right one position through the CY flag. The high order bit is set to the CY flag and the CY flag is set to the value shifted out of the low order bit. Only the CY flag is affected.

RAS Row Address Strobe. A signal used in dynamic RAMs to reduce the pin count by multiplexing the address. A group of pins is used at one point in time for one part of the address, then reused with different RAS to carry a different part of the address.

Raster Scan The most common technique of TV or CRT display. An image is created from groups of dots of varying brightness. See Vector Graphics.

RATFOR RAtional FORtran. A structured dialect of FORTRAN which is compiled into standard FORTRAN by a preprocessor.

R-C Resistor-Capacitor. A circuit connected to an oscillator to define its oscillating frequency. For stable frequencies, a crystal is required.

R<condition> 80C85 Assembly Language Instruction. Conditional return. If the specified condition is true, the actions specified in the RET instruction are performed; otherwise, control continues sequentially. The addressing mode is register indirect. No flags are set.

RD Received Data (RS-232C standard).

RDE Received Data Enable. A status flag in a UART.

RDOS Real-time Disk-Operating System. See DOS.

RDY ReaDY. A control signal used with slow memory or devices, to indicate that valid data is available.

READ BASIC Statement. Reads values from a DATA statement and assigns them to variables of the same type. The format is:

```
READ <variable list>
```

<variable list> is a list of variable names, numeric or string, which are to receive the values read from the corresponding DATA program line.

A READ statement must always be used with a DATA statement. The READ statement takes values read from the DATA statement and assigns them to READ variables on a one-to-one basis. The variable types must agree with the data being stored in them or a "?SN Error in <linnum>", a syntax error, occurs, where <linnum> is the line number of the DATA statement containing the type mismatch.

One READ statement can access several DATA statements, or several READ statements can access the same DATA statement. If the amount of data available is less than the number of variables allocated for the data, an out of data "?OD Error in <linnum>" error occurs, where <linnum> is the line number of the READ statement with the excess variables. If the number of variables in the READ statements is less than the number of DATA items, subsequent READ statements can be used and reading begins at the first unread element. If there are no subsequent READ statements, the extra data is ignored. Data can be reread using the RESTORE statement. See RESTORE.

READ Put external data into memory. Data in internal RAM is immediately available for processing. Data in external memory must be copied into internal memory, processed, then copied back out to external memory.

If data has been created in internal memory, it can be written out to external memory. If data read in from external memory has not been modified, there is no need to write it back out since the original copy is still there.

Read Character from Keyboard BASIC.

Keyboard Input Command Table	
Keyboard Input Command	Function Performed
INPUT	Prompts for data input from the keyboard
INPUT\$	Assigns a string of a given length, input from the keyboard, to a string variable
INKEY\$	Accepts the string value of the key currently pressed as variable data
LINE INPUT	Assigns a line of data, input from the keyboard, to a string variable
ON KEY GOSUB	Defines an interrupt subroutine to execute when a specific function key is pressed

Read Data from File BASIC. See File I/O Commands and Functions.

Read-Only Memory ROM. Storage that can be written only once. The Model 100 relies primarily on Random Access Memory (RAM), read-only memory (ROM), diskettes, and cassettes. Memory, by itself, is usually a reference to RAM. This is the general purpose, erasable and reusable memory located inside the Model 100.

ROM contains fixed data, usually programs such as the Model 100's BASIC ROMs and its ROM Operating System (Basic Input/Output System). The ROM contains the fundamental Machine language programs to run the various devices attached to the Model 100 such as monitor, printer, diskettes, cassettes, etc. ROM and RAM together make up the internal memory or main memory of the Model 100, or any other computer. See Memory.

Read/Write Describes the nature of an input/output operation, and the direction of data flow.

Real-Time An action, or a system capable of action, at a speed that keeps pace with the actual process.

Real-Time Operating System An operating system capable of real-time task management, including event scheduling, interrupt management, and real-time event counters.

Reasonableness Test A test to determine whether or not the value of a variable falls within a range defined as reasonable. It detects and filters noisy inputs or erroneous outputs.

Record A unit of information, either read, written, or stored, such as a punched card, a disk sector, or a line of characters.

Recursive A function, routine, or procedure which calls itself.

Redundancy Using more than one of the same item to increase reliability or performance.

Reentrant Programs or routines written in reentrant code. This code can be used by several tasks at the same time.

Reentrant Code A single segment of code and data which is not modified during execution, so that it can be called by multiple programs.

Refresh The logic necessary to rewrite the contents of the dynamic RAM periodically, typically once per millisecond (ms).

Refresh Circuitry Electronic circuitry which periodically reads and rewrites the contents of dynamic memory to prevent loss of data. See Dynamic Memory.

Register One word of memory, usually implemented in fast flip-flops, directly accessible to a processor. Most CPUs include a set of internal registers which can be accessed much faster than the main memory.

Register Select One or more lines used to select one register out of a given number within a device. Register select pins are normally connected to the address bus.

Relational Operators BASIC. See Operators.

Relative Addressing A method of memory addressing in which information is located by adding a displacement to a pointer. Addresses are expressed relative to some base address or pointer.

Release An edition of a software package. Programs and software packages are changed from time to time to correct errors or add new capabilities.

Relocatable • Reserved Words, Uses and Restrictions

ties. To keep programs from being in a constant state of flux and to simplify distributing modified programs to users, a number of changes are made, tested, and packaged as a new version or release of the program. See Version.

Relocatable The load module or object form of a program (or routine) which does not contain fixed addresses or which is structured so that it can be executed anywhere in the memory.

Rem A unit of radiation.

REM BASIC Statement. Used to preface non-executable comments or remarks in programs so that they are reproduced intact when the program is listed.

Remarks or explanatory comments in a BASIC program. The apostrophe (') can be used as a synonym for REM. The apostrophe can be used without a colon to indicate that the rest of the line is a remark. An example is:

```
20 REM Just a comment
30 ' Just a comment
40 LET X = 1: REM Just a comment
50 LET X = 1: ' Just a comment
60 LET X = 1' Just a comment
```

The colon (:) allows multiple BASIC statements on one line.

REM statements are nonexecutable statements and have no effect on the program when it is run, but are output exactly as entered when the program is listed. They do take up memory space and slow execution time. REM statements can be branched into by using GOTO or GOSUB statements. Execution continues with the first executable statement after the REM statement.

Remarks can also be added to the end of a line in which case each format requires different punctuation. If you add REM to the end of a program line, it must be separated from it by a colon (:). If you use the apostrophe (') to preface a <remark> it can be appended to the end of a program line with no punctuation needed.

Removable Media Physical modes of storing types of information, such as programs and data files are called media. The main media for the Model 100 is cassette, but RAM files and other devices are also used.

Media are classified as:

- a) removable media—such as diskettes, cassettes, and some hard disks, or
- b) fixed media—such as RAM files and most hard disks. Fixed media are not removable from the

device that drives them, so there is no ability to store additional data or backup copies off-line (outside the computing system) for insertion when needed. See Memory.

RENUM NEC PC-8201A BASIC Statement. RENUM rennumbers the lines in a program and changes all line number references in statements such as GOTO, GOSUB, THEN, ELSE, etc., to point to the new line number. The format is:

RENUM <(newnum)(,oldnum)(,increment)>

(newnum) is the line number that the numbering sequence is to begin with. 10 is the default.

(oldnum) is the line number where the renumbering sequence is to begin. If not specified, renumbering will begin with the first line of the program.

(increment) is the number used to increment the renumbering sequence. 10 is the default.

Renum will not assign line numbers greater than 65529 nor will it renumber programs in any way that would have the effect of changing the order of execution.

Renumber BASIC Program Lines See RENUM.

Repeat a Character BASIC. See STRING\$.

Repeat Program Lines BASIC. See FOR...NEXT.

RES RESet signal.

Reserved Device Names Names with special meanings in BASIC. They can be used in most places that a file name occurs, such as LOAD and SAVE statements. They name devices that can originate or receive data.

CAS: Cassette tape store file

COM: RS-232C transmission file

LCD: LCD screen display file

LPT: Printer display file

MDM: Modem transmission file

RAM: RAM memory stored file

See LOAD, LOADM, MERGE, OPEN, RUN, RUNM, SAVE, and SAVEM.

Reserved Word A specific value which serves a special purpose and can therefore not be used for other purposes. For example, NEW is a reserved word because it is used to erase current BASIC memory contents. See BASIC Reserved Words.

Reserved Words, Uses and Restrictions of Reserved words have particular meanings in BASIC and are used for commands, statements, and function and operator names. These words cannot be used as variable names and a variable name cannot be a reserved word followed by a type declaration

Reset • Resume a Program after a Pause

character (\$, %, !, #). Reserved words also should not be embedded within a variable as AND is embedded in HAND.

Delimiter (separate with space or spaces around them) reserved words so that they are easily recognized by BASIC. If you accidentally use one of the reserved words, you will get an error message or have strange results when you execute the program. See Debugging BASIC Programs.

Following are the reserved words in BASIC:

ABS	AND	ASC
ATN	BEEP	CALL
CDBL	CHR\$	CINT
CLEAR	CLOAD	CLOADM
CLOSE	CLS	COM
CONT	COS	CSAVE
CSNG	CSRLIN	DATA
DATE\$	DAY\$	DEF
DIM	DSKI\$	DSKO\$
EDIT	ELSE	END
EOF	EQV	ERL
ERR	ERROR	EXP
FIX	FILES	FOR
FRE	GOTO	GOSUB
HIMEM	IF	IMP
INKEY\$	INP	INPUT
INSTR	INT	IPL
KEY	KILL	LCOPY
LEFT\$	LEN	LET
LFILES	LINE	LIST
LLIST	LOC	LOF
LOAD	LOG	LPOS
LPRINT	MAX	MERGE
MDM	MENU	MID\$
MOD	MOTOR	NEXT
NAME	NEW	OFF
NOT	ON	OPEN
OR	OUT	PEEK
POKE	POS	POWER
PSET	PRESET	PRINT
READ	REM	RESTORE
RESUME	RETURN	RIGHT\$
RND	RUN	SAVE
SCREEN	SGN	SIN
SOUND	SPACE\$	STEP
SQR	STOP	STR\$
STRING\$	TAB(TAN
THEN	TIME\$	TO

USING
WIDTH

VAL
XOR

VARPTR

Reset Return either to zero, or to some arbitrarily selected beginning point. Pressing the reset button on the rear panel of the Model 100 performs a reset.

Reset Button The reset button is located on the rear panel of the Model 100 to the left of the RS-232C port. If the keyboard ever "freezes up" (doesn't respond to any key press) press the reset button to reset the system ("unfreeze" the keyboard) and return to the Main Menu. This does not effect the contents of your RAM files or the value of system variables such as TIME\$ and DAY\$.

Reside Where a program is recorded. Usually either cassette or in memory (RAM or ROM).

Resident A program which resides in the main memory of the system, either RAM or ROM.

Restart Press PAUSE twice to continue a paused program. Pressing the PAUSE command key once pauses the currently executing program.

RESTORE BASIC Statement. Rereads BASIC DATA statements from specified program lines. The format is:

RESTORE [<line>]

<line> is the line number of a DATA statement in the program.

Following the execution of the RESTORE statement with no line number, the next READ statement reads the first item in the program's first DATA statement. If a <line> is specified, the READ statement reads the first item in the DATA statement at the <line> number specified.

RESUME BASIC Statement. Continues program execution after an error recovery procedure has been performed or the SHIFT and BREAK/PAUSE keys have been pressed to stop BASIC program execution. The formats are:

RESUME
or
RESUME 0
or
RESUME NEXT
or
RESUME <line>

Resume a Program after a Pause Press the PAUSE command key a second time to continue a paused program. Pressing the PAUSE command key once pauses the currently executing program.

RET • Ring Indicator

The format used depends upon where program execution is to resume.

RESUME or **RESUME 0** resumes program execution at the statement that caused the error before.

RESUME NEXT resumes execution at the statement immediately after the one causing the error.

RESUME <line> resumes program execution at the specified <line> number.

RET 80C85 Assembly Language Instruction. **RETurn**. The content of the memory location whose address is specified in register SP is moved to the low-order eight bits of register PC. The content of the memory location whose address is one more than the content of register SP is moved to the high-order eight bits of register PC. The content of register SP is incremented by 2. The addressing mode is register indirect. No flags are set.

Retrofit Improve or change software or hardware by making additions.

RETURN BASIC Statement. Returns you from a subroutine to the line following the subroutine call. It is used with the **GOSUB**, **ON...GOSUB**, **ON TIME\$ GOSUB**, **ON MDM GOSUB**, **ON COM GOSUB**, and **ON KEY GOSUB** statements. The formats are:

RETURN

or

RETURN<linenum>

<linenum> is the number of the program line that you want execution to return to.

RETURN <linenum> allows non-local returns from the event trapping routines to a fixed line number while still eliminating the **GOSUB** entry the trap created. Be careful when using this non-local **RETURN** since any other **GOSUBs**, or **FORs** that were active at the time of the trap, still are active.

Reverse Video Also called inverse video. Some CRT terminals display dark characters on a light background as opposed to the standard light on dark. The Model 100 LCD normally displays dark characters on a light background. In inverse video the Model 100 CRTs displays light characters on a dark background; blocks of characters marked in **TEXT** mode are displayed in inverse video.

Reversi* Try to obtain the maximum number of squares as you compete against the computer or a friend. This board game challenges your strategic abilities. 24K; cassette. SilverWare.

RF Modulator A device which changes a composite video signal, required for most monitors, into a radio frequency signal for display on a standard television set.

RFI Radio-Frequency Interference.

RI Right In. The right input to a shifter. This is the pin-in for the incoming bit to fill the vacancy left by shifting all other bits in the word to the right.

Right Arrow →. The cursor is the underline symbol (or with some programs a different symbol such as a square) which appears on the screen to let you know where an action (such as entering, deleting, or inserting a character) takes place. The cursor moves to the right as you type and backs up to the left when you press the **DEL/BKSP** key. Pressing the cursor right key (→) causes this process to auto-repeat scrolling the cursor right along the current line. When the cursor reaches the rightmost position of the current line, it wraps around to the leftmost position of the next line to repeat the process. As long as the → key is held down, the cursor continues to move to the right and in the direction of the end of the file.

In **TEXT**, and in **TEXT** as invoked by the **BASIC EDIT** command, the → key performs additional functions if used in conjunction with the **SHIFT** or **CTRL** key. Pressing **SHIFT** and → moves the cursor to the first character of the next word to the right. Pressing **CTRL** and → moves the cursor to the rightmost character on the current line.

RIGHT\$ BASIC Function. Returns the specified number of rightmost characters from a given string. The format is:

<var\$> = RIGHT\$(<stringex>,<num>)

Right Justify The Model 100 text editor, **TEXT**, left justifies the text but does not right justify it.

<stringex> is the string expression that is the source of the characters returned by **RIGHT\$**.

<num> is an integer expression specifying the number of characters to be returned for the right end of the string expression <stringex>.

If <num> is greater than or equal to the length of <stringex>, then the entire source string <stringex> is returned. If <num> is zero, the null string (length zero) is returned.

See **MID\$** and **LEFT\$**.

RIM 80C85 Assembly Language Instruction. Read Interrupt Mask. The accumulator is loaded with the restart interrupt masks, any pending interrupts, and the contents of the serial input data line (**SID**). No flags are set.

Ring Indicator In telephone-based applications, such as telecommunications via modem, ring indicator is the signal on line which causes the bell to ring.

Ripple-Carry An addition technique where the carry coming out of an adder is propagated to the next adder. Carry look-ahead is a faster method.

Rise Time The time required to complete the low-to-high transition of a pulse.

RLC 80C85 Assembly Language Instruction. Rotate Left. The contents of the accumulator is rotated left one position. The low order bit and the CY flag are both set to the value shifted out of the high order bit position. Only the CY flag is set.

RND BASIC Function. Returns a pseudo-random number between 0 and 1. The format is:

$\langle v \rangle = \text{RND}(\langle \text{numex} \rangle)$

$\langle \text{numex} \rangle$ is a numeric expression that evaluates to a zero or non-zero value.

A non-zero value of $\langle \text{numex} \rangle$ causes a new random number value to be generated by the RND function. A zero value for $\langle \text{numex} \rangle$ causes the RND function to repeat the last random number it generated. Each time a program is run using RND, it generates the same sequence of pseudo-random numbers. If you want to enter the number sequence in a different place within the sequence each time you run the program you can access the value of TIME\$ to determine the entry point.

RO Right Out. The right output from a shifter.

Rollover Depressing two or more keys on a keyboard simultaneously. A good keyboard controller includes debouncing and multiple-key rollover protection.

ROM See Read-Only Memory.

ROMable Code which executes properly when placed in ROM memory. Segments of ROMable code have no temporary data storage areas and do not accept instruction modification.

ROTATE An instruction which shifts the contents of a register or word to the left or right. The bit coming in one end of the rotating word is generally the one falling off the other end; sometimes it is the old value of the carry bit (9-bit rotation).

Round-Robin A scheduling technique in which each task on a list cycles some necessary time on a repeating sequence. In round-robin scheduling each process or device corresponding to a task is guaranteed periodic service whatever the actual task traffic.

Routine A section of code written to perform an action, such as an input-character routine or a disk-write routine.

Row Scanning A technique which checks to find which key was pressed. Each row is scanned, and the output on the column is examined, resulting in identification of the key.

RPG Report Program Generator. A business-oriented programming language which uses a highly structured system of preformatted commands.

RPM Rotations Per Minute.

RPN Reverse Polish Notation. See Postfix.

RPROM Reprogrammable Read-Only Memory. See PROM.

RRC 80C85 Assembly Language Instruction. Rotate Right. The content of the accumulator is rotated right one position. The high-order bit and the CY flag are both set to the value shifted out of the low-order bit position. Only the CY flag is set.

RS Register Select. A control signal determining which of several eligible registers will be used in an MPU operation.

R-S Flip-Flop A flip-flop using two cross-coupled NAND gates. See Flip-Flop.

RST $\langle n \rangle$ 80C85 Assembly Language Instruction. Restart. The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. Control is transferred to the instruction whose address is eight times the content of NNN. The addressing mode is register indirect. No flags are set.

RS-232C The standard for connecting computer system components, such as modems or network data concentrators. It allows substantial variation as to what signals are passed. The Model 100 has an RS-232C interface on the rear panel of the computer. RS-232C communications can be controlled in Telcom mode and by using the BASIC statements ONCOM GOSUB, COM ON/OFF/STOP, LOAD, SAVE, and RUN.

RTOS Real-Time Operating System. An operating system in which data input and computer response proceed at the same rate.

RTS Ready To Send. An RS-232C standard signal.

RUN To run a BASIC program it must first be in BASIC memory or stored in a RAM file in tokenized BASIC format (.BA extension).

RUN • RUN a BASIC Program Automatically

To run a program in BASIC memory, enter BASIC mode and enter the BASIC command RUN. RUN can also be used to load and automatically run a BASIC file from another device.

To run a program stored in a BASIC format RAM file, place the cursor on the Main Menu over the name of the file and press ENTER.

To run a BASIC program that is not in a BASIC format RAM file or BASIC memory, you must load that program to BASIC memory from its current location (cassette, another computer, etc.) by entering BASIC and loading the program using the LOAD, CLOAD, or RUN commands. RUN automatically runs the program once it is loaded. LOAD and CLOAD also automatically run the program once it is loaded if you specify the ,R option. See RUN, LOAD, CLOAD.

RUN BASIC Command. Starts program execution. A device can be optionally specified, in which case the file given is first loaded from that device and then run. If no device is specified, the BASIC file currently in BASIC is run. The R option keeps open all currently open BASIC files rather than closing them and erasing other BASIC memory. ,R functions this way whether a new file is loaded and run or the existing one is run. The formats are:

```
RUN [<linnum>[,R]]  
or  
RUN "[<dev>:][<r>]<w> <p> <b> <s>"[,R]  
or  
RUN "[<dev>:]<filename>"[,R]
```

When using the first format, <linnum> refers to the program currently in BASIC memory and specifies the filename within that program where you want execution to begin. <linnum> is optional. RUN alone runs the program currently in BASIC memory beginning with the first line.

<dev> can be any of the following:

- RAM: loads a RAM file
- CAS: loads a cassette file
- COM: loads a communication file via the RS-232C
- MDM: loads a communication file via the built-in modem

If <dev> is not specified, BASIC assumes the device is RAM.

<filename> is the RAM: and CAS: device files which use the third RUN format requiring a <filename>. For a RAM file, a .DO (document) or a .BA (BASIC) extension can be included in the filename. For a cassette file, no extension should be used, just the six letter file name. RAM: and CAS: are both used for string files.

The second format uses <r> <w> <p> <s>, which are for communications files such as RS-232C (COM:) and modem (MDM:) files. These parameters should match those being used by the sending computer. Definitions and valid values for each parameter follow:

<r> is the baud rate. Valid values are listed below:

- 1=75
- 2=110
- 3=300
- 4=600
- 5=1200
- 6=2400
- 7=4800
- 8=9600
- 9=19200

If the communications file to be loaded and run is specified as coming from the modem (MDM:), then the <r> communications parameter should be omitted. The modem runs at 300 baud by default. If the default is altered, even by specifying the default value of 300 baud, the built-in modem is disabled.

<w> represents word-length in bits and can be set to values ranging between 6 and 8.

<p> represents parity and can be set to:

- E meaning even
- O meaning odd
- I meaning ignore
- N meaning none

<r> indicates stop bits, 1 for one stop bit and 2 for two stop bits.

<s> enables and disables XON/XOFF status. E enables and D disables it.

RUN a BASIC Program Automatically When you turn on the Model 100, the Main Menu Program automatically displays the menu screen and waits for an ENTER key press indicating which application program or RAM file you wish to use. If you position the cursor over the name of a RAM file containing an encoded BASIC program, indicated by the file extension .BA and press ENTER, the BASIC program autoruns. It is also possible to autorun a BASIC program rather than going to the Main Menu when the computer is first turned off, so that the chosen BASIC program will autorun the next time you Power on.

First, enter BASIC and write or load the program you want to autorun. Then save the BASIC program to RAM using your choice of legal file names, six characters or less beginning with a letter. Now key in the command:

IPL "filenm"

where `filenm` is the name you gave to the BASIC file to `autorun`, and turn off the computer. The next time you turn on your computer, the BASIC program will begin executing. See `IPL`.

Run a Machine Language Program Use the `LOADM` or `CLOADM` BASIC commands to load the program into BASIC high memory from its current location on cassette or in a RAM file. A portion of BASIC high memory can be protected from use by BASIC using the `CLEAR` command. When the Machine language program is loaded, the screen displays the start and end addresses the program was saved to in high memory and the entry address, if any. You can then run the program by using the BASIC `CALL` statement. See `CLOADM`, `LOADM`, and `CALL`.

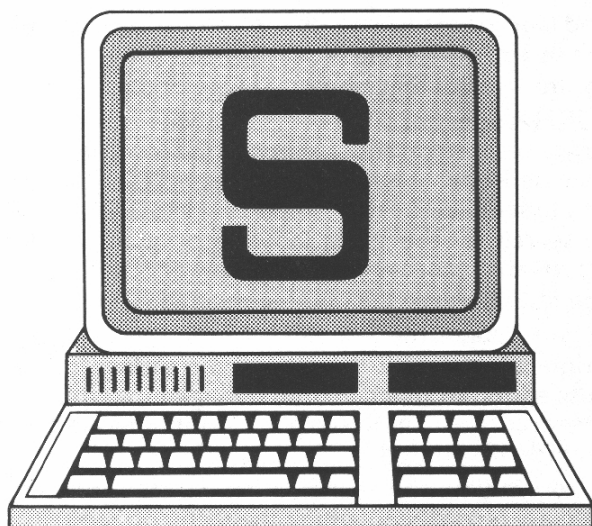
RUNM BASIC. Loads and executes a Machine language program stored as `<filename>`. The format is:

`RUNM ["<filename>"]`

`<filename>` is a string of up to six characters beginning with a letter. If you omit `<filename>`, BASIC loads and runs the first Machine language program it finds. See `Run a Machine Language Program`.

R/W Read/Write.

RZ Return To Zero. A recording technique often used on continuous magnetic media, especially tape.



S Codes. ASCII 83, HEX 53. s—ASCII 115, HEX 73.

S Select. Also, Strobe.

S-100 A common 100-line microcomputer system bus. It was originally developed as the bus for Altair, the first 8080-based system. It has since been standardized as the IEEE696 bus. In its latest form, it can be used with the newer 16-bit microprocessors.

Sample and Hold An analog circuit to capture and retain a signal so that it may be converted by an A/D converter.

Sampling Measuring an input value at intervals.

Satellite Processor A computer subordinate to another computer, possibly communicating over large distances, which performs specialized processing related to the master computer. The satellite may also be contained in the same device as the main processor, creating a multi-processor machine.

SAVE BASIC. To save a BASIC program named "SAMPLE" to RAM, enter:

SAVE "SAMPLE"

You can use F3, the Save key, to enter "SAVE" for you. If you are using a cassette, rewind it, press PLAY and RECORD together, and enter:

SAVE "CAS:SAMPLE"

To run the program later, use LOAD to copy it from the cassette or RAM file you SAVED it to back into BASIC. If you want to SAVE the program in ASCII format, which allows you to merge two programs, put ,A after the close quote of the program name. For example:

SAVE "SAMPLE",A

SAVE does not alter your program in memory. It is important to know that if you write a BASIC pro-

S • SBI <data>

gram, it is erased unless you SAVE it before you used the NEW command or LOADED a new program to BASIC. See MERGE, LOAD, and NEW.

SAVEM BASIC Command. Saves a Machine language program file, located between a given start and end address in memory, to cassette or RAM memory. The format is:

SAVEM "[<dev>:] <filename>, <startadd>,
<endadd> [,<entryadd>]

<dev> specifies the device that stores the Machine language file. Devices may be RAM: a RAM memory file, or CAS: a file stored on cassette tape.

<filename> is the name you are assigning to the file to be saved to cassette or to a RAM file. The memory address range defined by <startadd> and <endadd>. It can be up to six characters long and begin with a character.

<startadd> is the memory address that holds the first byte of data to be saved to a RAM or cassette file.

<endadd> is the memory address that holds the last byte of data to be saved to a RAM or cassette file.

<entryadd> is an optional specification. If included, it represents an entry address into the Machine language file other than the <startadd>. See Address Notation, and CSAVEM.

SBB M 80C85 Assembly Language Instruction. Subtract Memory with Borrow. The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

SBB <r> 80C85 Assembly Language Instruction. Subtract register with Borrow. The content of the second byte of the instruction is subtracted from the content of the accumulator. The result is placed in the accumulator. The addressing mode is immediate. Z, S, P, CY, and AC flags are set.

SBC Single Board Computer. A line of board-level products built to Intel specifications and using a common system bus known as the multibus. This bus is standardized as the IEEE 796. National Semiconductor has a compatible board line designated by the letters BLC (Board-Level Computer).

SBI <data> 80C85 Assembly Language Instruction. Subtract Immediate with Borrow. The contents of the second byte of the instruction and the contents of the CY flag are both subtracted from the accumulator. The result is placed in the accum-

ulator. The addressing mode is immediate. Z, S, P, CY, and AC flags are set.

SCCS Southern California Computer Society.

SCHEDL Mode A schedule-keeping program which selectively accesses previously recorded schedule information such as appointments, personal notes, and other reminders. To use SCHEDL you first create a file in TEXT named NOTE, containing the schedule information. Each entry in the NOTE file should be ended by pressing the ENTER key so that it constitutes one paragraph or record and is treated as a separate unit by SCHEDL. When TEXT saves the file to RAM it automatically gives it the extension .DO signifying that it is a text file. You can update the NOTE.DOC file at any time using the TEXT program.

When you enter SCHEDL, and ask for a string (symbol, word, or phrase), it searches all records in the NOTE.DOC file and displays all of the records that contain the string. If you try to enter the SCHEDL program without first creating a NOTE.DOC file the computer beeps and displays on the LCD:

"NOTE.DOC not found"

Press SPACE bar for MENU

To enter SCHEDL, position the cursor on the Main Menu over SCHEDL, and press ENTER. The prompt "SCHD:," appears to let you know you are in the SCHD: mode, and the eighth screen line displays the SCHEDL function key definitions. SCHEDL defines three of the six programmable function keys for use in the SCHEDL mode.

F1 finds and displays on the LCD all schedule records containing a string supplied by the user. If no string is given, F1 displays the entire contents of the ADRS.DOC file on the LCD, one screen at a time.

F5 finds and prints all records containing a string supplied by the user. If no string is given but ENTER is pressed, F5 prints the entire contents of the NOTE.DOC file on the printer.

F8 exits SCHEDL and returns to the Main Menu.

To find a record that contains a key string, press F1 or F5. This tells the program that you want to locate a string in the schedule file and displays all the records containing that string on the LCD (F1) or on an attached printer (F5). Whichever function keys you use, the program displays the prompt on the screen:

"SCHD:FIND"

Enter the string that you want to locate and press ENTER. You could look for a specific date or place. To easily retrieve all records of one type, designate symbols to represent different kinds of information

and label each record with the appropriate symbol when you create the NOTE.DOC file. Then search for the symbol that labels that type of file.

SCHEDL does not differentiate between upper and lower case letters and a search string is considered matched when it is located in a record even as part of a longer word. For example, if you used F1 or F5 to search for the string LET, the program would return any records containing the words letter and roulette because they both contain the string.

If you initiate the search using F5, the program prints any record containing the string, in its entirety, on an attached printer, beginning with the first record encountered and continuing until all records are printed.

If you initiate the search using F1, the program displays records containing the string on the LCD, beginning with the first record and pausing when the screen is full. Then if there are more records containing the search string, the program displays the prompts "MORE" and "QUIT" over function key markers 3 and 4. Pressing the F3 continues listing the files containing the search string until the screen is filled again, at which point, if more records remain, you are again prompted to press "MORE" or "QUIT." Pressing QUIT (F4) terminates the listing and returns you to the initial SCHD: prompt. To view or print all address records in the NOTE.DOC file, press F1 or F5 respectively, to enable the Find function, but do not specify a string. Instead just hit the ENTER key. The program lists all records to the LCD (F1) or the printer (F5) in the same way it lists some of the records when you specify a string to find.

Scheduling Allocating a non-shareable resource such as CPU time or an I/O device to a particular task for a period of time.

Schotky A technology of high-speed circuits.

SC/MP Simple Cost-effective MicroProcessor. National Semiconductor's 8-bit microprocessor.

Scope The scope of a variable or definition is that part of the program in which it may be accessed. Also, an abbreviation for oscilloscope.

Scotch A brand of magnetic recording media.

SCR Silicon Controlled Rectifier. A silicon cell which permits current flow in one direction only, thus converting AC current to DC.

Scratchpad A group of general purpose registers, without specific functions, that serve as a high speed workspace for some operations. Usually, it is an internal RAM, faster than the main system RAM.

Screen The surface of a monitor, TV set, or LCD on which characters are displayed.

Screen—ECHO Sends characters from the keyboard to the screen for visual confirmation of what has been typed. There is no hard-wired connection between the keyboard and the screen—the keyboard enters characters into memory. The ROM programs of the Model 100 copy the characters from memory to the screen, creating a duplication or “echo” of what was keyed.

Screen, Print PRINT copies the current display from the LCD screen to an attached printer, providing a dump of the entire screen.

Screen Display, Freeze or Hold To freeze the screen briefly during program execution from within the program, while the operator reads a message, write a delay loop after the message:

```
1000 FOR Y = 1 TO 2000
1010 NEXT Y
```

Or, put in a dummy input statement and instruct the operator to press ENTER. The input variable <A\$> need not be used in your program:

```
1000 INPUT "Press ENTER to continue";A$
```

Screen Down Scrolling You may scroll down text or program lines in a file, while in TEXT, BASIC EDIT, or Telcom terminal mode. In TEXT and BASIC EDIT, press the SHIFT and cursor down (↓) keys at the same time. If the cursor is already at the bottom of the screen, the next seven lines in the file scroll up onto the LCD screen. What was the eighth line becomes the first, with the cursor remaining at that line. It could be said that the screen is scrolling down the file, or the file is scrolling up the screen.

In Telcom terminal mode, you can view the eight previous lines of an incoming transmission file by pressing F1, the PREV or previous key. This scrolls them onto the LCD; pressing F1 returns you to the original screen. The effect is that of toggling your window or the “page” between the top and bottom halves of the text.

Screen Size A measure of the amount of information that a CRT screen can display. Screens can be measured diagonally, as TV sets, or by the number of vertical and horizontal dot or character positions. The Model 100 LCD screen has forty character positions per line and eight lines per screen. There are 240 pixel or dot positions per LCD line and sixty-four pixel positions per column.

Screen I/O BASIC Commands

LCD I/O Command	Function Performed
CLS	Turns off all LCD screen pixels
CSRLIN	Gives cursor's column position on the LCD
LCOPY	Prints current contents of the LCD to the printer
LINE	Draws a line or box on the LCD between coordinate points
POS	Gives cursor's line position on the LCD
PRESET	Turns off the LCD pixel at the screen coordinates given
PRINT	Prints data beginning at the current cursor position
PRINT@	Prints data at a specified LCD screen location
PRINT USING	Prints data on the LCD using a specified format
PSET	Turns on the LCD pixel at the screen coordinates given
SCREEN	Turns the function key label line (8) on and off
TAB	Tabs the cursor to the given LCD screen position

Screen Up, Scrolling See TEXT, EDIT, and Telcom terminal mode.

Scribe* This word processor for the Model 100 requires only 2K RAM to store and enhances the features in the TEXT program already built into your computer. Chattanooga Systems Associates.

Scrolling Moving the contents of the CRT screen up or down by one or more lines. Smaller movements, performed one dot at a time, are called microscrolling.

SDLC Synchronous Data Link Control. An IBM computer networking protocol.

SE Sign Extend. A technique used during a multiplication, division, and some shift operations to insure that negative numbers remain negative when shifted right. The convention is that bits shifted into the high end of the register are identical to the bit that was in the high order

Second Source • Seven Program Package*

position when the shift began. Also, Systems Engineer, a software technician usually employed by the computer manufacturer.

Second Source The manufacturer of a product, but not the original developer.

Sector A continuous section of a disk track. A block of data on a disk is addressed by its track and sector numbers. Typical disk sector sizes are 128, 256, or 512 bytes of data. Consecutively numbered sectors may or may not be physically adjacent within a track.

Sector, Bad A sector on the diskette which won't read/write data correctly, usually due to a minor physical flaw in the diskette. One or two bad sectors don't seriously affect the diskette's use; DOS marks them as bad and avoids using them. More than a few bad sectors indicate the diskette should probably be used as a frisbee.

Seek Time The time needed to position the read/write head in a diskette drive over the specified track of the disk.

Segment A continuous block of memory addresses, such as 0 to 64K.

Semicolon (;) When used in LPRINT to separate items, a semicolon means that the next item to print is immediately after this one, without a single space between. BASIC, however, brackets all positive numeric values with a blank on each side and all negative values with a leading unary minus and a trailing blank, whether you use a separator or not. The print zones begin in columns 0, 14, 28, 42, 56, and 70. See also Print Lines and Comma.

Send Line to Requesting Program Pressing the ENTER key sends the displayed line to the requesting program for processing.

Sensor A device which translates a physical stimulus into an electronic signal which may be used as computer input.

Sequencer In a bit-slice system, the module in charge of providing the next microprogram address to the microprogram memory. It is essentially a complex multiplexer, but may include stack facilities and a loop counter.

Sequential Access An access method in which items may be accessed in a fixed order only. The standard example of a sequentially-accessed medium is magnetic tape. Here, in order to access a particular record, all records before it must be scanned first. Cassette tape offers only sequential access.

Sequential File A file whose elements may only be accessed in ascending order. In order to read an element of a sequential file, all of the preceding elements must be accessed. RAM files on the Model 100 are sequential files.

Serial Data may be transferred between two computers or a computer and a peripheral in serial or parallel format. In most microcomputers, parallel I/O connections have eight wires to carry eight bits of a byte simultaneously (or in parallel). An example of a parallel interface is the printer interface. In a serial interface, though, only one data wire is available. In this case, the eight bits of a byte are transmitted one after another (or serially). The I/O device must collect all eight bits back into one eight bit byte. In the Model 100, the modem and RS-232C interfaces are serial.

Serial Data Data transmitted in sequence, one bit at a time.

Serial Port An I/O port through which data is transmitted and received serially. Serial ports are often used for communicating with terminals or other computers. Serial ports in the Model 100 are the cassette, RS-232C, modem, and the bar code interface.

Series Circuit elements connected so that the output of one is the input of the next.

Servo Short for servo-mechanism.

Servo-Mechanism A device responds to an electrical (or other) signal with a physical response, rather than producing another electrical signal. Servo-mechanisms range from a simple relay to robots.

Set the Date See DATE\$.

Set the Time See TIME\$.

Setup Time The time required before a signal can be changed from its prior state. Or, the time required to set up the preliminary conditions for a program to run—mounting required disks and tapes, changing printer forms, etc.

Seven Program Package* This set of seven programs features synchronized graphic and sound effects, and comes with audio instructions on cassette for each program. All the programs will work with 8K Model 100's. The programs are: "Bank Finance," with six functions including interest; "Sliding Numbers," a game for one or two players; "Vegas Slots;" "Bio-Rhythm," which shows the sine wave; "Calender;" "Vegas Blackjack," with a choice of dealer's hole card up or down; and

"Vegas Five Card Draw Poker." Complete Computer Services.

SGN BASIC Function. Gives the sign, positive, zero, or negative, of $\langle x \rangle$. The format is:

$$\langle v \rangle = \text{SGN}(\langle x \rangle)$$

$\langle x \rangle$ may be any numeric expression.

If $\langle x \rangle$ is positive, $\text{SGN} \langle x \rangle$ returns 1. If $\langle x \rangle$ is zero, $\text{SGN} \langle x \rangle$ returns 0. If $\langle x \rangle$ is negative, $\text{SGN} \langle x \rangle$ returns -1.

S/H Sample and Hold.

Shell The command interpreter running under the Unix operating system.

Shift Moving the contents of a register left or right by one bit or more. The bit falling out goes into the carry bit of the status register or is lost. The bit coming in is usually a 0, except in some special circumstances such as Sign-Extend.

SHIFT Key When the SHIFT key is pressed along with another keyboard key, it produces an upper case for alpha keys and alternate "upper case" definitions for most of the other keys. SHIFT does not effect the meaning of the function keys, the four command keys, or the ESC, TAB, CTRL, CAPS LOCK, NUM, and ENTER keys.

Shift Register A register whose contents can be moved left or right by one or more bit positions. See Shift.

Shifter A hardware device implementing the shift instruction. It moves all bits in a register left or right one bit.

SHLD $\langle \text{addr} \rangle$ 80C85 Assembly Language Instruction. Store H and L Direct. The content of register L is moved to the memory location whose address is specified in byte 2 and byte 3. The content of register H is moved to the succeeding memory location. The addressing mode is direct. No flags are set.

SI Serial Input. Also called sequential input.

Side Effect An unintentional change in the value of a global variable by a function, procedure, or subroutine. Structured programming languages discourage side effects by limiting the scope of global variables.

Sign Plus or minus. In two's complement notation, the sign can be determined by examining bit 7, the most significant bit (MSB).

Sign Magnitude A binary representation for integers where the MSB (most significant bit) acts as

the sign (0 for +, 1 for -) and the rest of the bits contain the magnitude, or absolute value, of the number.

Signed Binary A binary representation of signed integer numbers which sets aside one bit, usually the high-order or leftmost bit, to indicate the sign of the number.

Silicon Valley The area around Sunnyvale, in the Santa Clara Valley south of San Francisco, California, where many semiconductor manufacturers are located. This area contains the greatest concentration of electronics industries in the U.S. It is also called Silicon Gulch, but please, never Silicone Valley.

Silicon-Gate The MOS technology using silicon for the gate of the transistor. An alternative is aluminum-gate.

SIM 80C85 Assembly Language Instruction. Set Interrupt Masks. The contents of the accumulator are used in programming the restart interrupt masks. Bits 0-2 sets/resets the mask bit for RST 5.5, 6.5, 7.5 of the interrupt mask register, if bit 3 is 1 ("set"). Bit 3 is a "Mask Set Enable" control. Setting the mask (i.e. masked bit = 1) disables the corresponding interrupt. No flags are set.

Simplex Data transferred in one direction only.

Simulator A program which emulates a device by having the same input/output behavior as the device simulated. A CPU is easily simulated but I/O cannot be precisely simulated because of timing considerations, so only the logic of a program can be tested with a simulator.

SIN BASIC Function. Calculates and returns, in radians, the trigonometric sine function. The format is:

$$\langle v \rangle = \text{SIN}(\langle x \rangle)$$

$\langle x \rangle$ is the angle in radians.

To convert degrees to radians, multiply by $\text{PI}/180$; $\text{PI}=3.141593$.

Single-Board Computer A computer circuit board which contains all functions of the computer—CPU, ROM, RAM, and interfaces. Single board computers are often used for industrial control applications.

Single-Precision Arithmetic Regular arithmetic, or arithmetic done with single-word integers. See Double- or Multi-Precision Integers.

Single-Precision Variables Single-precision numbers (numbers with decimal fractions and up to six

Single-Sided • Slot

significant digits) can be declared by having their variable names end in I.

DEFSNG statement signifies that all variables following it which start with one of the specified letters are single-precision numbers. For example, if you place the statement: DEFSNG ABC at the beginning of your program, all variables starting with A,B, or C (such as Axis, Batch, and Check) are single-precision numbers because they start with one of the characters as a single-precision variable prefix for DEFSNG statements.

Variable names must start with a letter and can have any number of characters (although only the first two are significant). They must not use any reserved words such as: IF, ON, THEN, GOTO, etc., or a reserved word followed by a type declaration character (\$,% ,?,!,#) or a reserved word embedded within a variable name as RAT is embedded in FRAT. See BASIC Reserved Words, and BASIC Variable Names.

Single-Sided A method of disk storage using only one side of the disk. Also, a printed-circuit board with printed-circuit wiring on only one side.

Single-Stepping Executing a program one statement or step at a time. It is useful for debugging. See Debug or Test a Program.

Sink Current A logic family's current drive capability. Sink current is 1.6 millamperes for one standard transistor-transistor logic (TTL) gate.

SIP Single In-line Package. A package for a chip which has a single row of pins, usually very few in number (2-8). A dual-in-line package has up to 100 pins.

SIZE* Computes the total number of lines and bytes after reading a file. Checksum is also performed to verify telephone transmissions. Cassette. Micro Computer Services.

Size of Files BASIC provides no means of determining file size. File names and sizes are listed in the segment of the memory map that constitutes the Main Menu. Below is a BASIC program that reads this data from the memory map and displays it on the LCD:

```
10 REM File Directory with file lengths
20 LNE = 1
30 PRINT "[file][size] free";TAB(23)
40 FOR ETRY = 0 TO 154 STEP 11
42 TYPE =63919
50 LSB = 63920
60 MSB = 63921
70 FLNM = 63922
```

```
72 T$ = CHR$(PEEK(FLNM+STP+ETRY))
74 IF T$ = CHR$(0) THEN 170
80 FOR STP = 0 TO 5
90 X$ = CHR$(PEEK(FLNM+STP+ETRY))
92 IF X$ = CHR$(0) THEN X$ = " "
94 PRINT X$;
100 NEXT STP
110 PRINT ".";
120 FOR STP = 6 TO 7
130 X$ = CHR$(PEEK(FLNM+STP+ETRY))
132 IF X$ = CHR$(0) THEN X$ = " "
134 PRINT X$;
140 NEXT STP
150 PRINT (PEEK(LSB+ETRY)+
PEEK(MSB+ETRY)*256);
160 IF LNE=1 AND ETRY < 143 THEN PRINT
"";LNE=0 ELSE LNE=1:GOSUB 200:PRINT
TAB(23);
170 NEXT ETRY
172 IF P > 2 THEN 180
180 IF INKEY$ = "" THEN 180
190 END
200 IF ETRY = 11 THEN PRINT "space";
210 IF ETRY = 33 THEN PRINT "is";
220 IF ETRY = 55 THEN PRINT (FRE(0)+256);
230 RETURN
```

Sizes, Print See Type Formats.

Sizes, Type See Type Formats.

Skip An instruction to skip the following program instruction (usually in Assembly language). A condition is usually specified, such as:
"Skip If Z True"

Skip Over Print Zones See Double Comma and Print Zones, Spacing.

Skip to Top of Page See Form Feed and Pause.

Sky-Raider* An outer space action game featuring sound and graphics. Your task is to navigate your ship across the planet surface while waging battle with the deadly Borons. Alpine Data Systems.

Slave Any device imitating or under the control of another device.

Slew Rate A fast signal response measured in volts per second. Used in operational amplifier specifications.

Slice See Bit-Slice.

Slot A multiple-way connector allowing one printed-circuit board to be attached to another. Typically, an accessory board is plugged into a slot in the motherboard.

SLSI Super Large Scale Integration. A technology which holds up to 100,000 transistors per chip.

Small Scale Integration The technology which holds up to ten gates per device.

Smalltalk A language and software system developed by the Learning Research Group at the Xerox Palo Alto Research Center (PARC) during the years 1971-1980. It was released in 1981, and is organized around two fundamental concepts: objects and messages. Smalltalk systems are characterized by a high degree of pictorial interaction.

Small Talk* Software to interface the Model 100 to the IBM PC or the Commodore 64, enabling you to use the PC or C64 printer, disk, or screen for Model 100 data. Use it to send or retrieve files. A null modem is included. Key Solutions, Inc.

SMI Static Memory Interface.

Smoke Test Turning on the equipment for the first time to see if it operates.

SMS Scientific Micro Systems, a manufacturer.

SNOBOL StriNg-Oriented symBOLic Language. A character-string manipulation programming language.

SNR Signal to Noise Ratio.

SO Shift-Out bit. The bit lost (or stored in the carry status register) when a word is shifted left or right.

SOB Start-Of-Block. In TEXT and BASIC EDIT, you mark the beginning of a text block by pressing F7, the Select key. This marks the beginning of the block at the current cursor position. The end of the block is located at wherever you move the cursor.

Socket A mechanical electrical connector. The socket is also known as the female connector.

Soft-Fail Techniques which preserve a degree of system operation despite failures.

Soft-Sector A disk format where the beginning of every sector is detected by reading magnetic marks on the disk. In a hard-sectored disk, each sector's origin is marked by a hole.

Software Computer programs of all kinds are called software. Usually software is contrasted to hardware—the actual chips, wires, boards, etc. which make up the computer. A special case is Read-Only Memory (ROM), which is hardware that contains a permanent copy of software. The terminology here could be confusing, for “A BASIC ROM,” means a ROM (hardware) containing a

copy of a BASIC interpreter program (software). Such ROMs are often called “firmware” to distinguish them from non-program hardware and from software on media (diskette, cassette, etc.). The Model 100's built-in TEXT, TELCOM, ADDRSS, and SCHEDL programs are also ROM resident software. See Main Menu, TEXT, TELCOM, ADDRSS, SCHEDL, Hardware, and Firmware.

Software, Applications Programs that perform a function or group of related functions on the computer. These programs are called applications software packages when the programs are devoted to a task such as word processing, accounting, etc., and systems software packages when they facilitate the use of the machinery such as database management packages, disk operating system packages, or program development packages.

Software Package A pre-written group of commercially available programs designed to serve a specific need, such as word processing, inventory control, database management, etc.

Software-Compatible Describes CPUs which execute the same instructions—that is, speak the same Machine language.

Sort To arrange items according to defined criteria, such as alphabetical or numerical order.

SORT* Sorts data in alphabetic or numeric sequence. Listings are easily sorted into categories defined by the user. Cassette. Micro Computer Services.

SORT+* A quick utility program designed to run with the PUT+ file. Any of the sixteen categories can be accessed for the creation of a listing. Automatic alphabetization or numeric sequencing occurs with each category entry. This is part of BusinessPak+. 16K; cassette. Portable Computer Support Group.

SORT2+* Sorts even larger files than Sort+, but uses only 1K additional memory. Interacts with the DATA+ and PUT+ files for sorting data alphabetically or numerically. Each file quickly sorts its own records thus requiring only 1K memory extra. The user is prompted step-by-step in designating which field and file to perform this function. 16K; cassette. Portable Computer Support Group.

SOS Silicon-On-Sapphire. An integrated circuit technology in which a sapphire substrate is used to yield high operating speeds.

SOUND BASIC Command. Enables and disables the high pitched sound the computer makes when you load files from cassette or when the computer

SOUND • Source, The*

is waiting for the carrier signal in telecommunications. The format is:

SOUND ON

or

SOUND OFF

SOUND ON enables the above two sound generating conditions. SOUND OFF disables sound at those times. SOUND OFF does not disable sound generation in response to the BASIC BEEP and SOUND commands.

SOUND BASIC Statement. Used to generate sound through the speaker. The format is:

SOUND <pitch>,<length>

<pitch> is a numeric expression in the range 0 to 16383.

<length> is the desired duration in fiftieths of a second. It is a numeric expression in the range 0 to 255.

When SOUND produces a sound, the program continues to execute until another SOUND statement is reached. If the <length> of the new SOUND statement is zero, the current SOUND statement that is running is turned off. If the <length> is not zero (and in the valid range), the program waits until the first sound completes before it executes the new SOUND statement.

The following table contains notes and their <pitch> values for five octaves.

Octave

NOTE	1	2	3	4	5
G	12538	6269	3134	1567	783
G#	11836	5918	2959	1479	739
A	11172	5586	2793	1396	698
A#	10544	5272	2636	1318	659
B	9952	4976	2484	1244	622
C	9394	4697	2348	1174	587
C#	8866	4433	2216	1108	554
D	8368	4184	2092	1046	523
D#	7900	3950	1975	987	493
E	7456	3728	1864	932	466
F	7032	3516	1758	879	439
F#	6642	3321	1660	830	415

The duration for one beat can be calculated from beats-per-minute. Divide the beats-per-minute into 300, which is the number of <length> units per minute.

SOUND Generator I/O BASIC Command Table

If you find it necessary to signal the user with sound, refer to the BASIC command table in the next column.

Sound Generator Output Commands	Function Performed
BEEP	Makes the computer beep
SOUND	Makes the computer emit a tone of given pitch and duration
SOUND ON	Makes the computer emit a tone when loading a file from cassette and when waiting for the carrier signal in telecommunications
SOUND OFF	Disables SOUND ON function

Source The emitter of a transistor.

Source, The* The Source is a timesharing system that was founded in 1979 by Source Telecomputing Corporation. It provides the user with access to a large and varied number of programs and databases which are maintained in large computers in Virginia. In 1980, Source became a subsidiary of Reader's Digest and as of 1983 had over 37,000 subscribers.

The initial subscription to Source costs \$100.00. There are also hourly on-line fees that vary with the type of network used. Rates for the continental United States range from a daytime high of \$20.75 per hour if you log on between 7:00 am and 6:00 pm, Monday through Friday, to an evening low of \$7.75 per hour at 300 baud. 1200 baud service commands \$25.75 and \$10.75, respectively. Costs for Source Plus are about 1/3 higher and there is a monthly minimum charge whether you use the service or not. Transmissions can be carried over Telenet, Tymnet, and Uninet.

One very interesting feature is the storage capacity given to personal and business files on Source computers. One record equals 2,048 characters and costs from \$.50 per record per month down to \$.05 depending on volume (plus a minimum monthly maintenance and access fee).

The services offered by Source include current news and sports, educational programs, games, financial management, mailgrams, a public bulletin board, and access to hundreds of information sources. It also offers business databases, shopping by catalog, travel, and dining and entertainment information.

Source Plus features Media General Financial Service, Commodity News Service, abstracts from over thirty business publications, and Comp-U-Star, which allows you to purchase merchandise from an electronic discount catalog.

The Source encourages sharing of information through a public area where users set up a magazine that can earn them royalties, and a computer teleconferencing capability called PARTICIPATE.

Below are two sign-on procedures:

TYMNET SIGN ON

After dialing Tymnet you hear a high-pitched tone and see on the LCD screen:

“What sort of terminal are you using”?

Type in the terminal type ID code you were given when you received your account number:

SOURCE10;PRIM;

(substitute your system number if it is not 10). Press RETURN.

Type your ID number and press the space bar. Type your six-digit account number and press the space bar. Type your password. Press RETURN key.

You are now on Source and the system displays the Entry Screen of the Menu.

TELENET SIGN ON

Press the RETURN key twice.

Enter the identifier code for the type of terminal you are using, given to you when you received your account number. Press the RETURN key.

Type in the computer system code number given to you when you received your account number. Press the RETURN key.

There will be a display.

Type ID and press the space bar. Type your six-digit account number and press the space bar. Type your password. Press the RETURN key.

You are now on Source and the system displays the Entry Screen of the Menu.

You can also direct-dial Source if you live in a nearby calling area.

Many companies and professional organizations use the Source to create networks called Private Sectors for specialized information and communication services for their members. Special software and program guidelines have been developed to assist the creation of these networks. Source Telecomputing Corporation.

Source Code A synonym for source program.

Source Drive The diskette drive from which information/data is coming. The target drive is the diskette drive to which information/data is going.

Source Language The language on which a translator program operates to produce a version in Machine language.

Source Program A file which contains data to be processed by a language processor, interpreter, or compiler. A source program contains relatively readable statements in a language such as FORTRAN, COBOL, or BASIC. These were translated by a compiler or interpreter program, producing a Machine language program.

SP Stack Pointer.

Space In the RS-232C standard, the space, or binary 0, means negative voltage. If the binary 0 is used in a current loop, it means no circuit flow. When modeming, this designates the lower frequency of the pair. Also, commonly used as a synonym for the blank character. See Mark.

Space Up One Line To advance one line on the printer (space up) without carriage return, enter BASIC statement:

LPRINT CHR\$(10)

or use the “line feed” button (LF) on your printer. Entering only LPRINT gives a line feed—both a space up one line (line feed) and a return to left margin (carriage return).

Spaces, PRINT or LPRINT BASIC. See TAB.

SPACE\$ BASIC Function. Returns a string containing <num> spaces. The format is:

<v\$> = SPACE\$(<num>)

<num> is in the range 0 to 255.

Spacing BASIC Print Lines To get one or more spaces between fields printed by your BASIC programs, use a literal of spaces like: “ ”. To get several spaces between the printed values of A\$ and B\$, enter BASIC statement:

LPRINT A\$;“ ”;B\$

See Print Zones, Spacing.

SPDT Single Pole Double Throw. A type of switch.

Speaker BASIC. See BEEP and SOUND.

Special Characters, BASIC The following characters have special meanings in TRS-80 Model 100 BASIC. These characters cannot be used to have meanings other than those stated below:

- blank
- = equal sign or assignment symbol
- + plus sign or concatenation symbol
- minus sign
- “ double quotation mark or string delimiter
- / slash or division symbol
- \ backslash or integer division symbol

Speeding Up a BASIC Program • Stack

^	caret or exponential symbol
*	asterisk or multiplication symbol
%	percent sign or integer type declaration character
#	number (or pound) sign, or double precision type declaration character
\$	dollar sign or string type declaration character
!	exclamation point single-precision type declaration character
&	ampersand
,	comma
'	single quotation mark (apostrophe) or remark delimiter
.	period or decimal octal type declaration character
;	semicolon
:	colon or statement separator
?	question mark or PRINT abbreviation
—	underline
<	less than
>	greater than
(left parenthesis
)	right parenthesis

Speeding up a BASIC Program You can make your Model 100 BASIC programs run about twice as fast by defining the numeric variables to have the most efficient possible type. The default for numeric variables is double-precision, which is very slow and usually not needed. Use integer variables (DEFINT) for counters, indexes, etc. Decimal numbers requiring six or fewer significant figures (such as amounts up to 9999.99) should be single-precision (DEFSNG). The NEC PC-8201A uses single-precision variables by default, but using integers for counters and indexes speeds up NEC programs even more.

SPHL 80C85 Assembly Language Instruction. Move HL to SP. The contents of registers H and L (16 bits) are moved to register SP. The addressing mode is register. No flags are set.

Spikes Sharp, temporary increases in a signal or voltage.

Spinwriter A type of thimble printer manufactured by Nippon Electric Corporation (NEC).

Split-Screen Division of a display screen into two or more separate areas, or windows, in which distinct information is displayed.

Spool Simultaneous Peripheral Operating On-Line. A method of increasing system throughput by

allowing programs using slow output devices to complete execution rapidly. Program output data is placed in queues on high-speed mass storage devices or a part of main memory dedicated to spooling for low-speed transmission concurrent with normal system operation. See CSPOOL and MSPOOL.

Spreadsheet A program which quickly and efficiently performs all the traditional analysis calculations, such as cost projection, financial forecasting, investment analysis, budgeting, strategic planning, etc., making financial planning available to everyone. Spreadsheets have become invaluable tools in many businesses because they eliminate repetitious, time-consuming computations. Most spreadsheets divide the computer screen into a grid of rectangular cells and use coordinates to refer to the different cells. Creating a spreadsheet involves placing numeric values in cells and relating them to each other with mathematical formulas. For example, to perform an addition problem, cells are added together and the answer appears in another cell. See MultiPlan*, VisiCalc*, and PortaCalc*.

SPST Single Pole Single Throw. A type of switch.

SQR BASIC Statement. Returns the square root of x. The format is:

SQR(<num>)

<num> is any number.

Square Root BASIC. See SQR.

SR Status Register. A register holding bits that indicate the type of results that were obtained by the last operation, such as positive or negative, error or overflow situations.

SS Solid State.

SSDA Synchronous Serial Data Adaptor. A synchronous serial interface.

SSI Small Scale Integration. A technology holding a few gates per element.

SSR Solid State Relay.

STA <addr> 80C85 Assembly Language Instruction. STore Accumulator direct. The contents of the accumulator are moved to the memory location whose address is specified in bytes 2 and 3 of the instruction. The addressing mode is direct. No flags are set.

Stack A LIFO structure necessary for subroutine and interrupt management which reserves the chronological ordering of information.

Stack Pointer The register in the CPU which contains the address of the top of the stack in memory.

Stand-Alone A device which operates by itself, requiring no other equipment.

Start New Program NEW. Erasing BASIC memory. To start a new program type:

NEW

This completely erases all lines now in BASIC's memory. If BASIC's memory holds program lines you want to keep and you haven't already stored them on cassette or in RAM, SAVE it first. If you don't erase the program in memory before starting on another, you usually wind up with an unusable combination of lines from your old and new programs.

Start or Execute See RUN.

Start-bit A bit indicating the beginning of asynchronous serial transmission. See Stop-Bit.

Starting after a Pause Press the PAUSE command key a second time to continue a paused program. Press the PAUSE key once to pause in any program at any time. See Control Keys.

State Table A list of the outputs of a logic circuit based on the inputs and the previous outputs. Such a circuit has memory and cannot be described by a simple truth table. Also called state-transition table.

Statement A string of characters which is a syntactically complete instruction with respect to a high-level language translator.

Static Memory Called MOS memory, this uses a flip-flop as a storage element. It does not need to be refreshed and does not require a clock. It does not lose its contents as long as power is applied.

Static RAM RAM memory circuits which retain their contents as long as power is applied. See Batteries and Nicad.

Status The present condition of a device, usually indicated by flag flip-flops in special registers. See Flag.

Status Bit Handshaking The delegation of certain bits of a parallel I/O port to coordinate information transfer with a peripheral device. Status bits are used to indicate device read, buffer full, printer out of paper, etc.

Status Register A register used to hold status information inside a functional unit, such as an MPU, a PIC, a DMAC, or an FDC. A typical MPU status register provides carry, overflow, sign (nega-

tive), zero, and interrupt. It can also include parity, enable, and mask.

STAX <rp> 80C85 Assembly Language Instruction. Store accumulator indirect. The contents of register A are moved to the memory location whose address is in the register pair <rp>. Only register pairs <rp>= B (registers B and C) or <rp>= D (registers D and E) may be specified. The addressing mode is register indirect. No flags are set.

STC 80C85 Assembly Language Instruction. Set Carry. The CY flag is set to 1. No other flags are affected.

STD STanDard.

Stepper Motor A mechanical device which rotates a fixed amount each time it is pulsed. Often used in diskette drives.

Stock Market Software See Portafolio*.

STOP BASIC Statement. Terminates program execution, at some location other than the physical end of the program, and returns to command level. The format is:

STOP

You may use STOP statements anywhere in a program to stop execution. When BASIC sees a STOP statement, it displays the message:

Break in <linenum>

where <linenum> is the program line number where the STOP occurred.

STOP statements do not close files, as do END statements. BASIC always returns you to command level following execution of a STOP statement. You can use STOP statements while you debug your program so that you can examine or change variable values using direct mode and assignment statements. The CONT command resumes execution of the program directly after the program line where it was stopped as long as no BASIC program lines have been altered. See CONT and RESUME.

Stop Current Function To end a current function in a program at any time, press the SHIFT and BREAK/PAUSE keys together. To resume BASIC program execution after terminating in this way, enter the BASIC command CONT. This resumes execution from the point of the break. See Control Keys.

Stop-Bit A bit indicating the end of asynchronous serial transmission. The number of stop-bits is one of the communications parameters you set prior to using the modem or RS-232C interface.

Stop-Bits, Setting • String Variable

Stop-Bits, Setting Any time you transmit data between computers with the built-in modem or the RS-232C interface you must be sure that both computers are using the same communications parameters. Data transfers between computers can be initiated within TELCOM Entry mode or from BASIC using the LOAD, SAVE, RUN, MERGE, and OPEN commands.

The communications parameters are a series of one character values represented by the variable format `<r><w><p><s>`. Parameters and their values are:

`<r>` is the baud rate. Valid values are listed below:

- M = modem (300 baud)
- 1 = 75 baud
- 2 = 110 baud
- 3 = 300 baud
- 4 = 600 baud
- 5 = 1200 baud
- 6 = 2400 baud
- 7 = 4800 baud
- 8 = 9600 baud
- 9 = 19200 baud

If you plan to enable the modem with BASIC, you can omit the `<r>` value because the modem operates at 300 baud by default. To enable the modem within TELCOM specify M as the `<r>` value. If you specify any other `<r>` value the modem is disabled and the RS-232C baud rate is set to the value you selected.

`<w>` represents the word length in bits and may be set to values 6, 7, or 8.

`<p>` represents parity and may be set to:

- E = even parity
- O = odd parity
- I = ignore parity
- N = no parity

`` indicates stop-bits. 1 for one stop-bit and 2 for two stop-bits.

`<s>` enables and disables XON/XOFF status. E enables it and D disables it.

To set the communications parameters from TELCOM press F3, the Stat key. This displays the current default communications parameters (M711E,10 pps). The segment of the string before the comma contains the values corresponding to `<r>` `<w>` `` `<p>` `<s>`. The segment of the string after the comma represents the pulses per second that the modem will use to automatically dial the phone. You may change this value to 20 pps if your local phone system supports this. If you want to change any of the communications parameters, displayed by pressing the Stat key (F3), you must backspace

over and rekey the entire string. It is important to realize that TELCOM distinguishes between upper and lower case letters so use exactly the format given. It is not necessary to rekey the pulses per second specification if you are satisfied with the default value.

To set the communications parameters from within BASIC, refer to the formats given in the LOAD, SAVE, OPEN, and MERGE definitions in this encyclopedia. In each case, the BASIC command word includes the communications parameters in the `<r><w><p><s>` format and the same valid values apply (with the exception of M for the `<r>` option.)

Storage Synonym for Memory.

Strain Gauge A sensor which produces a voltage or resistance change when a force is applied.

Strike Method See Epson Printer.

STRING Functions in BASIC

String Function	Operation Performed
CHR\$	Gives character of ASCII code
INKEY\$	Gives keyboard key currently pressed
INPUT\$	Gives the characters from the keyboard
LEFT\$	Gives the left part of a string
RIGHT\$	Gives the right part of a string
SPACE\$	Gives a string constant of spaces
STR\$	Converts a number to a string value
STRING\$	Gives a character string

String An ordered sequence of data items, such as characters. For example, the word "string" is a string of six characters. String variables use one byte of memory for each character in the string. See Character String.

String, Length of BASIC. See LEN.

String, Numeric Value of BASIC. See VAL.

String Concatenation See Concatenating Strings.

String Graphics See Graphics, String.

String Handling The ability of a programming language to operate on strings of characters.

String Variable BASIC. Must end in \$ or start with a letter specified in a DEFSTR statement. String vari-

able names (for characters) must start with a letter and can have any number of characters, although only the first two are significant. They must not use any reserved words or a reserved word followed by a type declaration character (\$,% ,?,!,#). Also, reserved words may not be embedded as part of a variable name as LEN is embedded in LENGTH. String variables can hold 0 to 255 alphabetic, numeric, or special characters. See Reserved Words, Uses and Restrictions of, and BASIC Variable Names.

STRING\$ BASIC Function. Gives a string of length <len> whose characters all have ASCII code <num> or the first character of <str\$>. The formats are:

<var\$> = STRING\$(<len>,<num>)

or

<var\$> = STRING\$(<len>,<str\$>)

<len> and <num> must be in the range 0 to 255. <str\$> can be any string expression.

Strobe A selection signal that is active when data are correct on a bus.

Structured Language A computer language designed to aid or enforce structured programming. Control structures such as IF...THEN, ELSE, DO, WHILE, CASE and REPEAT...UNTIL, together with provisions for declaring logically separate program modules such as procedures, and limiting the scope of variables, all lend a modular structure to programs. Unconditional control transfer statements (GOTO) are often left unimplemented. Popular structured languages are Pascal, ALGOL, and C.

Structured Programming A set of techniques designed to increase the reliability and comprehensibility of programs by increasing programmer discipline. Structured programming involves precise problem specification, top-down or stepwise program design, and block-structured or modular programs.

STR\$ BASIC Function. Returns a string representation of the numeric value in <num>. The format is:

<v\$> = STR\$(<num>)

<num> is any numeric expression.

When <num> is positive, the string returned by STR\$ contains a leading blank, or the space reserved for the plus sign.

STTL See TTL.

STX Start of TeXt.

SUB M 80C85 Assembly Language Instruction. SUBtract Memory. The contents of the memory

location whose address is contained in the H and L registers are subtracted from the content of the accumulator. The result is placed in the accumulator. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

SUB <r> 80C85 Assembly Language Instruction. SUBtract Register. The contents of register <r> are subtracted from the contents of the accumulator. The result is placed in the accumulator. The addressing mode is register. Z, S, P, CY, and AC flags are set.

Sub-Harmonic A fractional multiple of the fundamental frequency.

Subroutine A program segment identified by a name and often bracketed by a "subroutine" and a "return" statement. Execution is transferred to a subroutine when a subroutine call occurs. Subroutines improve program modularity and save memory space at a minimal cost in the overhead required to process the call/return sequence. Subroutines may be called by BASIC statements ON...GOSUB, ON MDM GOSUB, ON TIME\$ GOSUB, ON COM GOSUB, ON KEY GOSUB, and GOSUB.

Subroutines, Passing Passing parameters (also called arguments or data) makes information from one process or program available to another process or program. The sender may be a program or a person typing the data onto a command line to be "passed" to a program. See CALL.

SUI <data> 80C85 Assembly Language Instruction. SUBtract Immediate. The contents of the second byte of the instruction is subtracted from the contents of the accumulator. The result is placed in the accumulator. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

SuperCalc* A versatile and enduring spreadsheet program for microcomputers. It is easy to use, and has the ability to output or retrieve data from other programs and databases. See MultiPlan*, PortaCalc*, Spreadsheet*, and VisiCalc*.

Support Chips All the components beyond the main device required for complete system operation.

SUT Socket Under Test.

Suzuki Hayashi One of the developers of the TRS-80. See Hayashi, Suzuki.

SW Status Word.

Switching Regulator A power supply design providing regulation by commuting the input voltage into a filter circuit.

SYBEX A leading publisher of computer books.

Symbol Table A table constructed by an Assembler or compiler to associate symbolic names with actual addresses or values.

Symbolic Describes the use of characters or character strings in a defined syntax to stand for machine-related entities such as instructions or data.

SYNC Short for synchronous or synchronizing.

Synchronous Operation controlled by a mutually sensed clock pulse.

Synchronous System A system in which all events are synchronized with a common clock pulse.

Syntax The rules governing proper construction of statements in a language. Punctuation, spelling and placement of keywords, data names, and number, type, and order of arguments. See BASIC Statements.

Syntax The set of grammatical rules defining valid constructs (sentences or statements) of a language.

Syntax Check A check performed by a program or person to ensure that one or more statements in a programming language complies with all syntax rules of the language. A program may pass all syntax checks and still give an error message or erroneous results due to logic, data, or program flow problems.

System Any aggregate of two or more interconnected electronic components. Also used for "computer system."

System Refers a group of programs which accomplish some function, a group of interconnected devices, or both.

System Diskette Contains DOS commands and other system software or utility-type programs. A system diskette can also store user programs and/or data. See also Data Diskette.

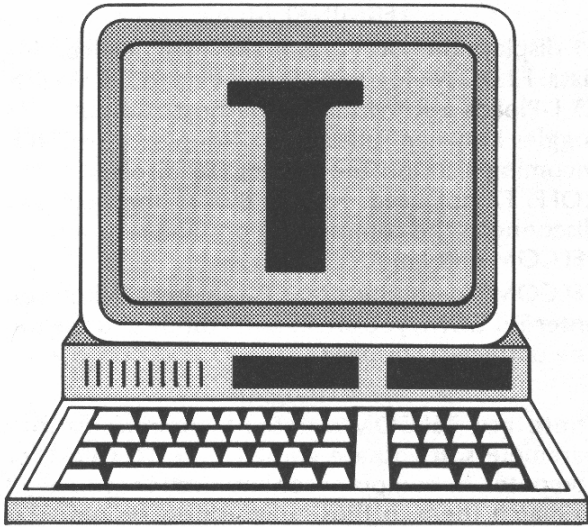
System Reset Press the reset button on the rear panel of the computer to reset the system when the keyboard will not respond.

System Time See Time\$.

System-Controller A chip such as the Intel 8228 chip used to de-multiplex the 8080's data bus into separate streams of data and control information. The 8085 and the Model 100's 80C85 have the system-controller integrated into the MPU chip.

Systems Software A software package is a group of computer programs (possibly including data files

and documentation) which perform a function or group of related functions on the computer. These are called systems software packages when they facilitate the use of the machinery—such as a database management package, a disk operating system, or a program development package. Programs are called applications software when the programs are devoted to an end user task such as word processing or accounting.



T Codes. ASCII 84, HEX 54. t—ASCII 116, HEX 74.

T An electrical network shaped like the letter “T.” It has one input, one output, and one ground lead. It is used with resistors for attenuators and capacitors, and inductors for filters. Also: True—a logical 1.

Tab When the key is pressed the cursor moves forward on the LCD eight spaces. This function is common in entering text.

TAB BASIC Function. Tabulates to position <num>. The format is:

```
PRINT TAB(<n>)
```

<num> must be in the range 0 to 255.

If the current print position is already past space <num>, then TAB goes to position <num> on the next line. Space 0 is the leftmost position. The rightmost position in a print row is 40 for the LCD display. For the printer it depends upon the default or defined line width. You are restricted to using TAB in the data list of an output statement such as PRINT or LPRINT.

Tabbing A method of moving a CRT cursor or printer head to a predefined column, called a tab stop, on the screen or paper.

Table Look-Up Method of converting one variable to a corresponding value or verifying its accuracy by searching a list or table of entries for the known keyword or value. The corresponding values may then be extracted from corresponding positions in the table. Looking up a name in the phone book to find the corresponding number is an example.

TAN BASIC Function. Gives the trigonometric tangent of <num>. The format is:

```
<var> = TAN(<num>)
```

<num> is the angle in radians. Multiply by pi/180 to convert degrees to radians. pi = 3.141593.

Tape, Installing See *Cassette Cable, Installing*.

Tape, Magnetic An inexpensive mass storage medium with the disadvantage of requiring sequential access. It is convenient for large files, or archival storage. Often the only external storage on very low priced systems and used as a backup for disks on larger systems.

Target Drive The diskette drive to which information/data is going. The source drive is that from which information/data is coming.

Task A particular execution of a program.

TBMT Transmitter Buffer eMpTy. One of the five status bits of a standard UART. It indicates true when a buffer may be reloaded. See PE, FE, OR, DAV.

TCAM TeleCommunications Access Method (IBM mainframe telecommunication term).

TD Transmitted Data (RS-232C standard).

TDM Time-Division Multiple access. A networking technique used in timesharing that regulates the flow between terminals.

TELCOM A ROM resident telecommunications program that has two modes of operation. The first, entry mode, is used to initiate telecommunications between two computers. Once a phone link or direct connection is established, terminal mode communication is established through the built-in modem or the RS-232C interface. Either an acoustic coupler and modem cable, or a direct connect modem cable attach the computer to the phone lines. TELCOM will autodial numbers to autolog onto a host computer system or information network, using information stored in an ADRS.DO text file you create in TEXT mode. Generally speaking, TELCOM uses the built-in modem along with a modem connector to transmit files. However, you can choose to use the RS-232C interface instead while you are in entry mode.

TELCOM Entry Mode: When you first enter TELCOM, the program is automatically in entry mode. Entry mode helps establish a communications link in three ways: by retrieving a phone number or other access information from the ADRS.DO file; by selecting the parameters and the device to be used in the communication; and, if used with the direct connect modem cable, by automatically establishing the phone connection, autodialing,

TELCOM

switching to terminal mode, and logging on to a host system, if you set up your ADRS.DO entry to do that. You can also bypass the phone lines and the modem to send data directly to another computer attached to the Model 100's RS-232C port.

TELCOM Terminal Mode. By the time you enter terminal mode you should have already used entry mode to set the parameters and device specifications for communications. You may or may not have automatically completed a log on sequence in entry mode. This mode lets you receive data from the host system which is displayed on the Model 100 LCD, send files and information from the keyboard, print all incoming data on the printer, save incoming information in a RAM file, and terminate communications. To use terminal mode, the ANS/ORIG switch on the left side panel of the computer must be set to ORIG.

Before using TELCOM, you need to connect the built-in modem to a telephone or connect the RS-232C port to another computer. TELCOM automatically uses the modem unless you tell it otherwise in entry mode. The acoustic coupler has the advantage of working with any kind of phone but also has the disadvantage of not supporting TELCOM's autodial feature. Additionally, the connection may be noisier than the direct connection provided by the modem. If you are using the acoustic coupler, you must set the DIR/ACP switch on the left side panel of the computer to ACP. The modem cable can only be used with a modular telephone, not, for instance, with most one piece public telephones. When using the modem cable, you need to set the DIR/ACP switch to DIR. If you choose to use the RS-232C interface the modem is automatically disabled and a new set of coupling devices must be specified.

Function Keys in TELCOM. TELCOM uses the programmable function keys to initiate various program activities. The two TELCOM modes, Entry and Terminal, each define these keys differently. A description of their functions are in each mode as follows:

ENTRY MODE

F1 finds and retrieves a phone number from the ADRS.DO file and temporarily changes the meaning of F3 and F4:

F3 displays MORE phone numbers for the same name. F4 QUITs the Find.

F2 CALLs the phone number on the LCD. F3 changes the STATUS of TELCOM communications parameters. F4 enters TERMInal mode. F5 not used. F6 not used. F7 not used. F8 exits TELCOM Terminal mode and returns to the Main Menu.

TERMINAL MODE

F1 displays the PREVIOUS eight lines of incoming data. F2 DOWNloads incoming data to a RAM file. F3 UPlods a RAM file to the host computer. F4 toggles between full- and half-duplex. F5 ECHOs incoming data on the printer. F6 displays XON/XOFF. F7 not used. F8 says Bye to the host and disconnects the communications link returning to TELCOM entry mode.

TELCOM Communications Parameters. When you enter TELCOM, you are in entry mode and the top line of the LCD displays:

M711E,10pps

These are TELCOM's default communications parameters. If you are going to use TELCOM for computer-to-computer communications you need to match them to those of the host system. The parameter characters before the comma represent baud rate, word length, parity, stop bit, and status. The characters after the comma represent the pulse dial rate. For many of the information networks, the preset values will already be correct. To change any one of the values, press F3, the status key, and reenter the line; then press ENTER. The program distinguishes between capital and lower case letters. A brief description of each communications parameter and its possible values follows.

Baud Rate. A measurement of data flow in bits per second. The TELCOM default value is M, or 300 bps, which signifies that the modem will be used. If you change the baud rate from M to some other value, even 300, the modem will be disabled and you set the baud rate for the RS-232C interface. Other acceptable values for the baud rate follow:

TELCOM Value	Baud Rate	Device	Used With
M	300	modem	
1	75	RS-232C	rarely used
2	110	RS-232C	do teletypewriters
3	300	RS-232C	slow-speed serial printer
4	600	RS-232C	printers and disc devices
5	1200	RS-232C	high speed serial printer; printers, some modems
6	2400	RS-232C	direct computer to computer
7	4800	RS-232C	"
8	9600	RS-232C	"
9	19200	RS-232C	

Word Length. The number of bits that represent a character. The default value is seven bits. You may use 6, 7, or 8 bits as the word length but seven- and eight-bit words are most common.

Parity. A means of detecting if bits have been lost in a file transfer. All bit arrays are made even or odd before a transfer and checked to see if they are the same after a transfer. Possible values for parity are:

- I—ignore parity
- O—odd parity
- E—even parity
- N—no parity

The default value in TELCOM is I for ignore parity although no parity (N) is also frequently used in telecommunications.

Stop-Bit. Special character or characters that signal the end of a binary word as it is transmitted in telecommunications. You may specify 1 or 2 stop-bits. The most common and the default value is 1 stop-bit.

Line Status. Causes a temporary pause on the sending end if the receiving end is having difficulty keeping up with the incoming data. The receiving system sends an XOFF signal to the sender to pause transmission and an XON signal to the sender to resume transmission once the backlog of information has been handled by the receiving system. These signals are recognized automatically if this feature is enabled on the Model 100. E enables this feature and D disables it. The default is E.

To manually send an XON from the Model 100 keyboard to another computer, press CTRL and Q together. To send an XON, press CTRL and S.

Pulse Rate. This is not really a communications parameter. It is an instruction telling the Model 100 modem how many pulses per second (pps) to use when pulse-dialing the telephone. All phone systems should work at the rate of 20pps. You must reenter all other values if you change any communications parameter. You don't need to re-enter the pulse rate if you are happy with the default value.

Finding a Phone Number. After creating an address file called ADRS.DO in TEXT, TELCOM uses that file to locate and retrieve phone numbers for manual or automatic dialing. To do this, ADRS.DO file names should have the name first, then a phone number enclosed in colons. Enter TELCOM by positioning the cursor over TELCOM on the Main Menu and press ENTER. This displays the current telecommunications parameters on the LCD along with the prompt telcom:. To retrieve a phone number, press F1, the Find key, and enter the name of the person you want to call. The program goes to the ADRS.DO file and looks for the first occurrence

of that name in the file. If the name is not found, the prompt is repeated. If the name is found, the program displays it and the phone number that follows it. Any portion of the address entry following the second colon in the phone number sequence is not displayed. You can press F3, the More key, after a find is initiated to search for the next occurrence of the person's name. This is particularly useful if you have two different numbers for one person; however, they must be in separate entries for this to work. If you decide that you want to find a different name and number, press F4, the Quit key.

Autodialing a Phone Number. Once a name and phone number is displayed on the LCD (using F1) in TELCOM entry mode, you have the choice of dialing it automatically or manually. To autodial, you need to have the Model 100 connected to a telephone with the modem cable. Then press F2, the Call key, and the message "Calling:" is displayed, followed by the person's name. As the number is dialed, the digits appear on screen. Pick up the receiver before all the digits are dialed if you want to talk to the person on the other end. There is no need to pick up the receiver if you are calling another computer. Press F4, the Terminal key, to switch the computer over to terminal mode, unless you have set the phone number sequence in the address entry to do it for you automatically. Remember to set the ANS/ORIG switch on the left side panel of the computer to ORIG before entering terminal mode.

Manually Dialing a Number—Acoustic Coupler. An acoustic coupler which doesn't support autodialing for telecommunications between computers. Requires manual dialing. Use the Find function to locate and display the number on the LCD and then dial the number yourself on the telephone. When the connection is established for telecommunications with the host system, set the ANS/ORIG switch to ORIG, attach the coupler cups to the telephone receiver, and press F4 to enter terminal mode. See Acoustic Coupler.

Manually Dialing a Phone Number—Modem Cable. There are some instances when you can't autodial a number. For instance, if you have not created an ADRS.DO file, you can't use it to retrieve a number for dialing. In this form of manual dialing, the phone must be connected to the modem cable. Enter TELCOM and press F2, the Call key. The prompt "call:" is displayed and you enter the number you want dialed. Use only the numbers—not dashes or other separators. The message "calling:" is displayed, followed by the number being called. Pick up the receiver before dialing is com-

plete if you want to speak to another person. If you are calling another computer, set the ANS/ORIG switch on the left side panel of the computer to ORIG and press F4 to enter terminal mode. You can also lift the receiver and dial the number on the telephone while it is connected to the modem, replacing the receiver when you are safely in terminal mode.

Entering Terminal Mode. Terminal Mode allows the Model 100 to transmit and receive data through phone lines or with another computer. Before entering Terminal Mode, set the ANS/ORIG on the left side panel of the computer to ORIG and the telecommunications parameters used by Telcom to those used by the other computer. After either dialing the number of the host computer, or connecting the RS-232C ports, enter Terminal Mode manually by pressing F4, the terminal key. As the computer enters terminal mode it makes a high pitched sound and displays a new set of function key definitions on the eighth line of the LCD. If you are using the modem cable to autodial the host computer, you automatically enter Terminal Mode by including the symbols ≠ following the phone number but before the second colon. For instance, the ADRS.DO entry for the host computer might look like this:

Home Base :3312395≠: Headlands, CA

If you were to find this entry (F1) and had the Model 100 call it for you (F2), once the modem dialed the number Telcom would automatically go to the terminal mode. Avoid using the ≠ symbols in an address entry you access and autodial for person to person conversation, because leaving terminal mode and returning to entry mode for conversational purposes disconnects the phone.

In Terminal Mode incoming information is displayed on the LCD. You can back up to view the last screenful by pressing F1, the Previous key, in TELCOM Terminal Mode. Pressing F1 again takes you back down the "page" to the eight lines you saw before.

While in Terminal Mode you can generate a record of all data received by printing it on an attached printer by pressing F5, the Echo key.

Download a File. In TELCOM Terminal Mode, incoming telecommunications data may be saved, or downloaded, into a RAM file for later use by pressing F2, the Down key. This prompt appears:

File to download?

The program is asking for the name of the file that will hold the incoming information. As with any other RAM file name, the first character must be a letter and you may use a maximum of six charac-

ters. After entering the file name, press ENTER to begin downloading. The down label above the F2 function key marker appears in reverse video during the download. The incoming data is displayed as it downloads to memory. To stop downloading, press F2 a second time. If the information you are downloading fills RAM memory before it is finished, downloading automatically terminates, retaining whatever portion of the file was received up to that point. After exiting TELCOM mode you can use TEXT to access and manipulate the contents of the file you have downloaded.

Full-/Half-Duplex. You can send data to the host computer by entering it on the keyboard or by uploading a previously created TEXT file. In either case, the characters are displayed as they are transmitted. The meaning of the characters depends upon whether you are using full- or half-duplex. Full-duplex echoes the characters that the host computer has received on the LCD so you can see if your data was altered in transmission, due to a "noisy" phone line, for instance. Half-duplex, on the other hand, echoes the characters you send on the LCD and provides no way for you to tell if they were transmitted intact. In TELCOM terminal mode F4, the Full key, lets you toggle between full- and half-duplex.

Upload a File. Allows you to send a previously created RAM file to the host computer. For instance, if you use TEXT to write a report while you are traveling, you can call the computer at your home office and upload the file. Press F3, the up key, and when the prompt "file to upload?" appears, enter the name of the file you want to send and press ENTER. TELCOM responds by displaying the prompt "width?". You then enter a line width between 10 and 132 characters-per-line and press ENTER. The default width is 40 characters-per-line. TELCOM uses this information to insert carriage returns into the data it sends, formatting the page width for the host system and preventing split words. This is essential in some cases. For instance, CompuServe cannot handle data input at more than 132 characters without a carriage return. While uploading the function key label for F3, UP appears in reverse video. Be careful, because if you press any keys during an upload that character is inserted into the file you are sending.

Ending Telecommunications. F8, the Bye key, terminates transmission. TELCOM double checks with you by displaying the prompt "disconnect?". Enter Y for yes and N for no. Pressing F8 a second time takes you back to the Main Menu.

See Acoustic Coupler; and Modem Cable, Installing.

Tele-stock* Manage your stock portfolio, analyze your investments, and retrieve market activity on selected stocks from the Dow Jones News Service at a set time each business day. This data automatically updates your history files which in turn provide the data for graphic display of the high, low, volume, and close for each stock. This software also calculates return on investments and supports both the acoustic coupler and modem cable for interfacing with communications lines. 24K. Telesoft.

Teletype One of the oldest and slowest peripherals for communication with a computer.

TELEX+* An easy-to-use program for telephoning or mailing information across the country or world. A modem is used with the computer for communication purposes. This method of electronic communications can greatly reduce telephone costs for the average business person. 16K; cassette. Portable Computer Support Group.

Tenkey+* Your Model 100 numeric keypad becomes a super-calculator when you use this software. It features three memories and continuous onscreen display of their contents. It also has several very handy built-in functions, such as future value, internal rate of return, amortization, effective interest rate, present value, and depreciation. Portable Computer Support Group.

Terminal Emulators See AST-327X, and Model AST-3780.

Terminal Mode A computer's CRT and/or printer can be used as a terminal for another computer. The Model 100 TELCOM program operates in both Entry and Terminal Modes. Terminal Mode is accessed from TELCOM Entry by pressing F4, the Term key. See Emulation, TELCOM.

Terminate Current Function Press the SHIFT and BREAK/PAUSE keys together. See Control Keys

Test Data, Test Run A programmer must insure that a program will correctly process all types of data. Samples of the data are prepared (test data) and the program is executed using this data (a test run). The program's outputs (reports, screen displays, files, etc.) are then verified. An error in the processing logic of a program is called a bug, hence the terms "debug" and "bug-free." See Debug.

Test Sites, Alpha and Beta Two stages of testing is done by companies to make sure that their programs are usable. Alpha testing is usually done in-house, before sending the product out for Beta testing, by companies who know that their product is incomplete or has flaws.

Beta test sites are generally larger in number. They try a product under the assumption that it is essentially complete and correct. If they discover errors in the program they tell the originators. The companies ordinarily attempt to fix the errors rapidly so the Beta sites can stay "on the air" with the new product. If numerous or serious bugs are found, the product may have to go back to Alpha testing until an improved version can be presented for another round of Beta testing.

Text Editor An editor program primarily for text files. A text editor manipulates ASCII characters such as letters and punctuation marks. TEXT is the text editor in the Model 100 and is permanently in ROM. It creates and changes letters, reports, programs or telecommunications files. In BASIC EDIT, the line or range of lines you specify are translated from tokenized BASIC to ASCII format. BASIC EDIT enables all the text editing functions of the TEXT program. See TEXT Mode.

Text File A file containing character data, letters, numbers, or special characters. These files are always assigned a .DO extension when saved into RAM memory. See Data File.

Text Mode The Model 100 TEXT mode is a ROM resident text processing program. It has many functions you need to create and edit text documents, such as those used by the ADDRESS and SCHED programs, as well as other files for your own use.

Opening and Closing Text Files. To enter TEXT mode, move the cursor on the Main Menu over TEXT and press ENTER. This is displayed on the screen:

File to edit?

Type in file name no more than six characters long and press ENTER. If there is a TEXT file in RAM that already had the name you gave, the program loads it and displays the first eight lines on the LCD. If it is a new file name, TEXT creates a new file. Pressing F8 exits TEXT, and automatically saves the file to RAM with the file extension .DO, which signifies that it is an ASCII or document file. The file name is displayed on the Main Menu. A shorter method for opening a previously created TEXT file is to move the cursor to the file name on the Main Menu and press ENTER.

Creating a TEXT File. As you enter text on the keyboard, the cursor moves to the right and the text you type appears. If the length of the text exceeds forty characters, the cursor moves down to the next line. This is called an automatic text wrap feature. The only time you need to press the RETURN or ENTER key is when you want to end a unit of text

Text Mode

such as a paragraph. This is equivalent to a line feed and sends the cursor to the first position on the next line. As you enter text and fill the screen, the top line moves off the top of the screen as a new line appears at the bottom. This is known as scrolling.

Moving the Cursor. You can move the cursor without affecting the text by using the four cursor movement keys at the upper right of the keyboard. Pressing a cursor key once moves the cursor one space in the direction indicated by the arrow next to the key. If you hold the key down, the cursor continues moving. In moving left and right, the cursor wraps in the same way it does when you are entering text, to the next line. Likewise, when the cursor reaches the top or bottom of the screen, new lines scroll onto the screen. This is called continuous scrolling. Pressing SHIFT and a cursor UP or DOWN key moves the cursor to the top or bottom of the screen. If the cursor is already at the top or bottom, the next screenful scrolls onto the LCD. Pressing CTRL key cursor UP or DOWN moves the cursor to the top or bottom of the file. Pressing SHIFT and a cursor RIGHT key moves the cursor to the first character of the next word to the right of the current word or, if it is already there, to the first character of the next word to the right. Pressing a cursor right or left key and the CTRL key at the same time moves the cursor to the right or left end of the current line.

Inserting and Deleting Characters. To insert text, place the cursor where you want to insert the character and begin entering text. The new text is inserted to the left of the cursor position. To delete an unwanted character, place the cursor to the right of the character and press the DEL/BKSP key. Or, place the cursor under the unwanted character and press SHIFT and DEL/BKSP. As you insert or delete text, the characters after the cursor move as you make the change to fill in or make room for the spaces created.

Defining TEXT Blocks. If a large rewrite is necessary, changing one character at a time can be tedious. Fortunately, TEXT mode allows you to define a segment of the text as a whole block which can be manipulated. You can erase, move, or copy the block. To define text as a block, move the cursor over the first character and press F7, the Select key in TEXT. Next, move the cursor to the last character you want in the block. All the text in between the two positions appears in inverse video, grey on black, rather than black on grey, to indicate that it is a block.

Manipulating TEXT Blocks. To delete or move the

block, use F6, the Cut key, in text. To make a copy of the block, use F5, the Copy key. In both cases, the block is copied to an internal storage location, called the paste buffer. If you chose Cut, the text below fills in the space the block occupied. In either case, you can insert the block of text in the paste buffer anywhere in your text file, or even into another file. Do this by placing the cursor where you want the insert to go and pressing the PASTE key. The text after the cursor moves over and down to make room. Once you place a block of text in the paste buffer, it remains there until you replace it with another block or cold start the computer.

Finding a String. The Find function scans the file to find any string you specify. Because Find only searches from the current cursor position to the end of a file, you should move the cursor to the top of the file before initiating a find. Find also considers upper and lower case versions of the same letter to be identical. Press F1, the Find key, and the prompt "String" appears in the lower left corner of the LCD. Enter the string to be found and press ENTER. If the string is not found, the message NO MATCH appears on the screen. If the string is found, it appears on the LCD. To search for the same string again, press F1, reenter the string, and press ENTER.

Printing a TEXT File. If you have a printer attached to your computer which is turned on, press PRINT to print the screen contents, including function key labels. To print the contents of the entire file, press the SHIFT and PRINT keys together. The prompt "width" appears followed by the last print width used. The width default is 40. You can change the value to any width between 10 and 132, by backspacing and reentering the value. Once the width is set, press ENTER. To stop printing before the file is finished, press the SHIFT and BREAK/PAUSE keys together.

Saving the File on Cassette. To save a file to an attached cassette recorder, press F3, the Save key. This displays the prompt "Save to" on the screen. Enter the name of your cassette file, and make sure that the tape is ready to record. Pressing ENTER begins the recording process and you will hear a high pitched sound and see the tape move. When the copy is complete, the "Save to" prompt disappears.

Loading a Cassette File to TEXT. Connect the cassette recorder and position the tape at the beginning of the file you want to load or at the beginning of the tape if you don't know the file location. You can either add the contents of the cassette file to the end of an existing text file or

create a new text file. Enter the file name where the incoming data will be stored and press F2. The prompt "Load from" is displayed. Enter the file name from the cassette and press ENTER. You hear a high pitched sound and the tape moves. A "Skip" message and the names of files not matching the one you want appears on the screen. FOUND; and the file name that you requested appears when the file is located. In this case, the file name is the same as the one you gave for the "load from" prompt. When the file is loaded, the prompts disappear and you are ready to use the file.

Deleting a TEXT File. TEXT does not provide a means of erasing files. You need to enter BASIC. Do this by going to the Main Menu, placing the cursor over BASIC and pressing ENTER. The prompt "OK" appears. Enter:

KILL "file.do"

where "file.do" is the name of file you want to erase. If you need a reminder of what that name is, you can get a listing of all RAM files from BASIC by entering FILES and pressing ENTER.

Function Keys in TEXT. Some of the function key definitions are the same as those for other Model 100 applications programs, others are unique to TEXT. Below are the values of the function keys in TEXT mode.

F1 FIND a string.

F2 LOAD a text file from cassette.

F3 SAVE a text file to cassette.

F4 not used.

F5 COPY a marked text block to the paste buffer.

F6 CUT a marked text block saving a copy in the paste buffer.

F7 SELECT the beginning of a text block to mark for cut or copy.

F8 Exit the text processor and return to Main MENU.

Pressing the LABEL key once in TEXT displays labels corresponding to the capitalized portion of the above descriptions on the eighth line of the LCD. Pressing LABEL a second time turns off the labels and opens the eighth line for text processing.

Control Keys in TEXT. Most of the functions performed in TEXT by the cursor and programmable function keys can also be performed by a combination of the CTRL and alphabet keys. The following table shows TEXT control codes, their function, cursor and other special key equivalents and the functions.

Printer Control Codes in TEXT. In TEXT, you can embed printer control codes in a file that will underline your text. All printer control codes must be preceded by a CTRL-C, which displays on the LCD as ^P. As you see in the Control table you see that if the ^O and the ^N aren't preceded by a ^P their meaning is changed; it becomes the same as hitting the F5, Copy key, and the F1, Find key, in TEXT. To turn on underlining, follow the ^P with a CTRL-O which displays as ^O. To turn the underlining off, use ^P followed by CTRL-N which displays as ^N. A file containing these embedded printer control codes must be printed in a special way for the codes to work. If it was printed in the usual way, using the PRINT key, or SHIFT/PRINT combination, the control characters would be printed just as they appear on the LCD. To translate them to underlining, print the file with a version of the SAVE process. Press F3, the Save key, but use LPT instead of a file name, the device name for the printer. This "saves" the file to the printer, which prints it. Unfortunately, this way of printing does not provide a line length specification. You have to format your own text and provide carriage returns.

ASCII Code	CONTROL Key Combination	TEXT Key(s) Equivalent	TEXT Function Performed
1	CTRL A	← SHIFT	Moves cursor to start of first word to left
2	CTRL B	← SHIFT	Moves cursor to last LCD line or last line of next screen if there already
3	CTRL C	SHIFT BREAK/PAUSE	Cancels FIND, SAVE, LOAD, PRINT or SELECT function
4	CTRL D	→	Moves cursor right one character
5	CTRL E	↑	Moves cursor up one line on LCD
6	CTRL F	→ SHIFT	Moves cursor to start of first word to right
7	CTRL G	F3	Saves current file
8	CTRL H	DEL BKSP	Deletes character to left of cursor

Thimble Printer • Token Values in BASIC

ASCII Code	CONTROL Key Combination	TEXT Key(s) Equivalent	TEXT Function Performed
9	CTRL I	TAB	Moves cursor to the next TAB stop word to right
10	CTRL J	————	Not used
11	CTRL K	————	Not used
12	CTRL L	F7	Mark the beginning of a text block
13	CTRL M	RETURN	End a paragraph and carriage return
14	CTRL N	F1	Find a string (if after CTRL R stop underline printer control)
15	CTRL O	F5	Copy text block to paste buffer (if after CTRL P start underline printer control)
16	CTRL P	————	Preface printer control character
17	CTRL Q	CTRL ←	Moves cursor to far left of current line
18	CTRL R	CTRL →	Moves cursor to far right of current line
19	CTRL S	←	Moves cursor left one char
20	CTRL T	↑ SHIFT	Moves cursor to top LCD line or top of the next screen if there
21	CTRL U	F6	Delete a block of text, copy it to paste buffer
22	CTRL V	F2	Load a cassette file to TEXT
23	CTRL W	CTRL ↑	Moves cursor to first file character
24	CTRL X	↓	Moves cursor down one line on LCD
25	CTRL Y	PRINT SHIFT	Prints the entire file on printer
26	CTRL Z	CTRL ↓	Moves cursor to the last file character

Thimble Printer Prints fully-formed characters with a daisy wheel type element, but with the type petals bent up 90 degrees to form a cup or thimble shape.

TIME\$ BASIC. TIME\$ may be used to set or display the system time. To set TIME\$, enter BASIC and use the BASIC TIME\$ statement.

TIME\$ holds an eight character string variable of the form hh:mm:ss. This is a two-digit representation for each hour, minute, and second values, separated by a colon (:). Values for the hour (hh) range between zero (00) and 23. Values for the minute (mm) range between zero (00) and 59. Values for the second range between zero (00) and 59.

To set the system time, use TIME\$ in an assignment statement. The format is:

TIME\$ = "hh:mm:ss"

Two digits must be entered for each time data seg-

ment. Use a leading zero for one digit values. After you enter the system time, the Model 100 operating system automatically updates it to keep it current, even when the computer is turned off. This is true as long as the rechargeable, built-in, nicad battery continues to supply power to the RAM memory.

To display the system time, use TIME\$ as a variable. For example:

```
OK
PRINT TIME$
12:10:25
OK
```

The time is displayed in the upper left hand portion of the Main Menu screen. The ON TIME\$ GOSUB interrupt uses the same system value of TIME\$ that you set and displayed using the TIME\$ statement.

Token Values in BASIC Tokenized BASIC is a compressed form of storage for BASIC files. It sub-

Token Values in BASIC

stitutes 1-byte codes for BASIC keywords rather than storing each letter of the keyword as an ASCII value. Token values for the BASIC keywords are listed in the following table.

BASIC Keyword	Token Value
ABS	225
AND	213
ASC	249
ATN	237
BEEP	177
CALL	185
CDBL	244
CHR\$	250
CINT	242
CLEAR	167
CLOAD	168
CLOSE	154
CLS	176
COM	173
CONT	164
COS	234
CSAVE	169
CSNG	243
CSRLIN	202
DATA	131
DATE\$	171
DAY\$	172
DEF	161
DSKI\$	200
DSKO\$	152
DIM	133
EDIT	147
ELSE	145
END	128
EOF	239
EQV	216
ERL	196
ERR	197
ERROR	148
EXP	233
FILES	157
FIX	245
FRE	226
FOR	129
GOSUB	140
GOTO	136
HIMEM	204

BASIC Keyword	Token Value
IF	138
IMP	217
INKEY\$	201
INP	227
INPUT	132
INSTR	199
INT	224
IPL	187
KEY	175
KILL	189
LCOPY	179
LEFT\$	252
LEN	246
LET	135
LFILES	159
LINE	146
LIST	165
LLIST	166
LOAD	155
LOC	240
LOF	241
LOG	232
LPOS	228
LPRINT	160
MAX	183
MDM	174
MENU	186
MERGE	156
MID\$	254
MOD	218
MOTOR	182
NAME	188
NEW	191
NEXT	130
NOT	206
OFF	203
ON	151
OPEN	153
OR	214
OUT	150
PEEK	238
POKE	162
PRESET	181
PRINT	163
PSET	180
POS	229

Transistor • Truth Table

BASIC Keyword	Token Value
POWER	184
READ	134
REM	142
RESTORE	139
RESUME	149
RETURN	141
RIGHT\$	253
RND	231
RUN	137
SAVE	158
SCREEN	190
SGN	223
SIN	235
SOUND	178
SPACE\$	251
SQR	230
STEP	207
STOP	143
STRING\$	198
TAB(192
TAN	236
THEN	205
TIMES\$	170
TO	193
USING	194
VAL	248
VARPTR	195
WIDTH	144
XOR	215
+	208
-	209
*	210
/	211
^	212
\	219
>	220
=	221
<	222
'	255

Transistor An electronic device which can use one electrical signal to influence another. Transistors are used mainly as amplifiers and switches. Used as an amplifier, a transistor uses the changes in a small signal to make large changes in a large signal. Used as a switch, the transistor opens or

closes a circuit depending on the state of a controlling signal.

Traveling Appointment Manager, The* Scheduling and rescheduling of appointments and dates are easily managed automatically. The information is entered for later retrieval by screen or printer. This is part of The Business Management Series. Cassette. Traveling Software, Inc.

Traveling Communicator*, The Links the office or home computer with the TRS-100. Transmission of data occurs via telephone lines or direct cable connection. The Business Management Series. Cassette. Traveling Software, Inc.

Traveling Expense Manager*, The Records expenses, updates checking and credit card accounts, and displays reports. The Business Management Series. Cassette. Traveling Software, Inc.

Traveling Tax Manager*, The Simulates 1040 tax forms. It greatly decreases this year-end chore by performing the calculations after receiving the correct information. Registered owners are sent four quarterly tax newsletters for the latest tax updates. The Business Management Series. Cassette. Traveling Software, Inc.

Traveling Time Manager*, The A billing system which tabulates the total number of hours worked, standard billing rates, and the number of hours of equipment usage. The Business Management Series. Cassette. Traveling Software, Inc.

Traveling Accountant, The* Records and processes financial expenditures generated away from the home and office. The information obtained may be sent to the computer system at the office. Several different reports are available for production of financial statements. This is part of The Business Management Series. Cassette. Traveling Software, Inc.

Tree Structure A collection of data organized so that each item is linked to other items, creating a spreading network of linkages analogous to the branches of a tree. An example of data organized in this way might be the parts list of an airplane. The main entry is for the entire airplane. It is linked to the major components, such as wings, fuselage, and tail. Wings, in turn, are linked to their smaller components, such as flaps, engine supports, engines, etc. Each of these can be further linked to smaller and smaller subassemblies, until individual parts, such as nuts and bolts, are reached.

Truth Table A table showing the logical value (true or false) of a compound logical expression,

based on the logical value of the simple components of the expression. Example:

A	B	A or B
T	T	T
T	F	T
F	T	T
F	F	F

This table reflects the definition of OR: A or B is true if and only if either A or B or both are true.

TTL Transistor-Transistor Logic. Logic circuits based on bipolar devices, usually low-power Schottky circuits. These are fast but expensive because gold-plated Schottky diodes are required on every TTL bus input.

Tutor+* Learn touch typing while you have fun playing this version of Space Invaders. Instead of using a joystick, you move the firing gun by typing in words and phrases. Portable Computer Support Group.

Two-Pass See Pass.

Type Declaration Characters and Statements BASIC uses four kinds of variables to store four kinds of data. They are integer numeric, single-precision numeric, double-precision numeric, and string variables. The three numeric data types are interchangeable because data can be assigned from a variable of one numeric type to a variable of another numeric type. For arithmetic operations, all variables are temporarily stored and operated upon in double-precision numeric format and then stored back to their respective variables in the format specified by the numeric variable type. String and numeric variable data is not interchangeable in this way.

The default data type used by BASIC when none is

specified is double-precision numeric. One way to specify a variable that is not double-precision numeric by default is to append a type declaration character to the end of the variable name. This is a one character value that defines what type of variable data may be assigned to that variable name. The four variable type declaration characters used in BASIC are: (See table below)

Type Declaration Statements are another way to change the default variable type. Use the first letter in the variable name as an argument in a BASIC type definition statement such as:

DEFINT for integer variables
DEFSNG for single-precision integers
DEFDBL for double-precision integers
DEFSTR for string variables

For example, to specify string variables using the definition statement, the format is:

DESTB B, L, T

This defines all variables with names beginning in B, L, or T (such as BACON, BEEF, LIGHT, TOT, and THINK) as string variables. The BASIC type definitions will not affect the type declarations character tag (% , ! , # , \$) values. For example, TOT% would remain an integer variable and resist definition as a string variable.

Type Formats BASIC. To set the Epson RX-80 printer's print size, strike method, or lines-per-inch, you must send control codes to the printer. Enter BASIC and input an LPRINT statement containing ASCII control characters and other special codes. For example: (See end of Ts)

Combinations of the five type sizes with five strike methods produce twenty-five different type formats. Each type format can be printed at any of three standard spacings.

Finer control of line spacing is also available by

Character	Data Type	Data Description
%	Integer Numeric	A whole number value ranging between -32708 and 32767. Requires 2 bytes of memory to store.
!	Single Precision Numeric	A decimal number with six significant digits. May use exponential notation (preface exponent with an E). Requires 4 bytes memory to store.
#	Double Precision Numeric (default variable type)	A decimal number with 14 significant digits may use exponential notation (preface exponent with D). Requires 8 bytes of memory to store.
\$	String Variable	A sequence of 0-255 characters (numeric, alphabetic, special characters) enclosed in double quotes. Each character in a string variable requires one byte to store.

Types of Files, Cassette • Type+*

using dot spacing. This also allows double- and triple-spacing. The Epson printer prints 72 dots-per-inch.

You can specify the number of dots to be used as a print line. Twelve dots-per-line is the default, which is activated as soon as the printer is turned on. Twelve dots by 1/72 inch-per-dot = 12/72 inch, or 1/6 inch-per-line, or 6 lines-per-inch.

Two other standard dot settings are 9 dots-per-line: 9 dots by 1/72 inch/dot = 9/72 inch, or 1/8 inch, or, 8 lines-per-inch;

and 7 dots by 1/72 inch/dot = 7/72 inch/line, or 72/7 lines per inch (10.29).

You can specify <n>/72 lines-per-inch and <n>16 lines-per-inch, where <n> ranges between 0 and 85 and 0 and 255, respectively.

Since 12 dots-per-line makes 6 lines-per-inch, single-spaced, 24 dots-per-line gives double-spacing, 36 dots-per-inch gives triple-spacing, etc. The following table gives common values that you can use to verify your calculations:

LPI	Dots/Line	ASCII Control Characters Required	Alternate
7/72	single 7	27,65,7,27,50	27,50
	double 14	27,65,14,27,50	None
	triple 21	27,65,21,27,50	None
8	single 9	27,65,9,27,50	27,48
	double 18	27,65,18,27,50	None
	triple 27	27,65,27,27,50	None
6	single 12	27,65,12,27,50	27,50
	double 24	27,65,24,27,50	None
	triple 36	27,65,36,27,50	None

Any number of dots-per-line from 1 to 85 can be specified. Using the 27,65 in control character sequence redefines the default of 12 dots-per-line. Any subsequent use of the 27,50 "return to default" option returns to this new default, not 12(= 6 LPI). To reset the standard default of 12 dots-per-line, use 27,65,12,27,50. Entering 27,65,n does not change the line spacing; it only changes the stored default line spacing. The 27,50 moves the stored default line spacing into the current line spacing slot, making it immediately effective.

Using 27,48 or 27,49 changes the current line spacing, but does not affect the stored default line setting. A 27,50 control sequence returns to whatever

you last specified as the stored default line spacing—or to the 12 dots-per-line stored at power on if you haven't changed the line spacing.

To set the printer for 8 lines-per-inch, double-spaced, use the following sequence of control characters, from the table above:

27,65,18,27,50

To send these to the printer from BASIC, enter:

```
LPRINT CHR$(27);CHR$(65);CHR$(18);  
CHR$(27);CHR$(50)
```

Remember that you can reset all of the printer's specifications to the default values by turning it off and back on.

Some programs do not allow you to set the printer values from within the program. In these cases, you should set the printer before executing the program. This does not allow you to change print formats within a document, though it will recognize embedded print commands.

Another technique is to embed the control characters in the text. Some programs will recognize these; others won't.

For example, to print large, very dark characters at 8 lines-per-inch use double-width, emphasized, and double-strike all together by entering the following BASIC statements:

```
LPRINT CHR$(27);CHR$(14);'Turn on  
double-width
```

```
LPRINT CHR$(27);"E";'Turn on emphasized  
LPRINT CHR$(27);"G";'Turn on double-strike  
LPRINT CHR$(27);"O";'8 lines-per-inch
```

Anything printed after these printer control LPRINTs are printed in double-width, emphasized, double-strike characters.

To return to standard printing, turn off each of the non-standard print options:

```
LPRINT CHR$(20);'Turn off  
double-width
```

```
LPRINT CHR$(27);"F";'Turn off emphasized  
LPRINT CHR$(27);CHR$;"H";'Turn off  
double-strike
```

Multiple printer control characters can be given in one LPRINT, so:

```
LPRINT CHR$(20);CHR$(27);"F";  
CHR$(27);CHR$;"H";
```

performs the same function as the three LPRINTs listed above.

Types of Files, Cassette See Cassette.

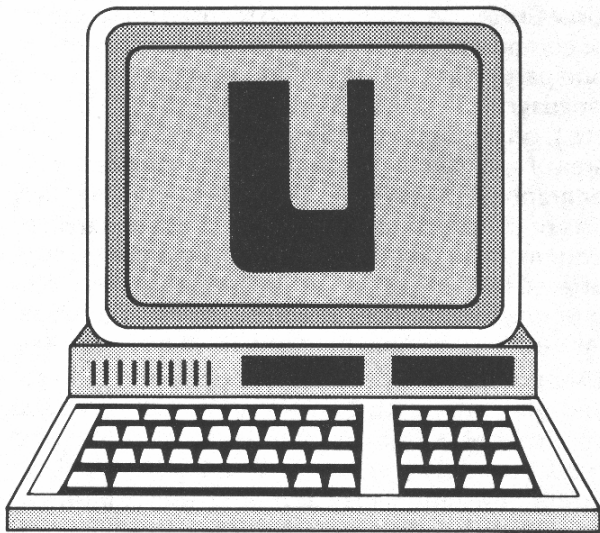
Types of Files, RAM See RAM.

Type+* Turn your Model 100 and an attached printer into a typewriter. Features a variable-size

Type Formats—continued

Print Type	Characters Per Inch	To Turn On (Enter BASIC Line)	To Turn Off (Enter BASIC Line)
Pica	10	LPRINT CHR\$(27);“P”;	Default value
Elite	12	LPRINT CHR\$(27);“M”;	LPRINT CHR\$(27);“P”;
Enlarged	5	LPRINT CHR\$(27);CHR\$(14);	LPRINT CHR\$ (20);
Condensed	17	LPRINT CHR\$(27);CHR\$(15);	LPRINT CHR\$ (18);
Strike Method			
Print Type	To Turn On (Enter BASIC Line)	To Turn Off (Enter BASIC Line)	
Emphasized	LPRINT CHR\$(27);“E”;	LPRINT CHR\$(27);“F”;	
Double Strike	LPRINT CHR\$(27);“G”;	LPRINT CHR\$(27);“H”;	
Italics	LPRINT CHR\$(27);“4”;	LPRINT CHR\$(27);“5”;	
Underline	LPRINT CHR\$(27);“-”; CHR\$(<i><n></i>) (<i><n></i> is 1 or 49)	LPRINT CHR\$(27);“-”; CHR\$(<i><n></i>) (<i><n></i> is 0 or 48)	
Line Spacing			
Line Spacing	To Turn On (Enter BASIC Line)	To Turn Off (Enter BASIC Line)	
1/6 Inch	LPRINT CHR\$(27);“2”;	Default Value	
1/8 Inch	LPRINT CHR\$(27);“0”;	LPRINT CHR\$(27);“2”;	
7/72 Inch	LPRINT CHR\$(27);“1”;	LPRINT CHR\$(27);“2”;	
<i><n></i> /72 Inch	LPRINT CHR\$(27);“A”; CHR\$(<i><n></i>) <i><n></i> ranges from 0 to 85	LPRINT CHR\$(27);“2”;	
<i><n></i> /216 Inch	LPRINT CHR\$(27);“3”; CHR\$(<i><n></i>) <i><n></i> ranges from 0 to 255	LPRINT CHR\$(27);“2”;	

buffer that allows you to correct mistakes on-screen, set tabs and margins, and center your text automatically. You can also use this software to print more than one copy of your text. Portable Computer Support Group.



U Codes. ASCII 85, HEX 55. u—ASCII 117, HEX 75.

U Underflow. A lower case u sometimes represents the Greek letter mu, meaning micro.

UART Universal Asynchronous Receiver/Transmitter. A serial-to-parallel and parallel-to-serial converter. Usually a particular kind of integrated circuit used to interface a byte-parallel port to a bit-serial communications network or processor bus.

UCSD p-System An integrated program development system created by Kenneth Bowles at the University of California at San Diego. It includes an operating system, a full-screen text editor, and compilers for Pascal, FORTRAN, and BASIC. Although originally developed to teach programming, it has been enhanced to provide the tools necessary for large-scale programming projects.

The p refers to an imaginary “pseudo” computer. The system compilers produce a very compact p-code, which runs on a “pseudo” computer. An interpreter converts the p code into acceptable code for the computer which runs the program. This makes the system easy to move between computers, or portable. Only a very small interpreter need be written for each computer on which the p-System runs. The p-System now works on over fifty different computers, with new ones being added all the time.

The system is highly interactive and consists of five main parts:

- 1) The full-screen Editor allows you to easily write programs and text. The Advanced Screen Editor has editing capabilities equal to sophisticated word processors.

U • Upload a Communications File

- 2) Compilers convert your source code into p-code. The Pascal compiler is integrated with the Editor, and when a syntax error occurs, you can return to the Editor at the point of error. The Compiler error messages are accurate and informative.

- 3) The Filer allows you to format disks, and to list, copy, rename, and remove files. File names may be up to sixteen characters long. The Filer also checks and marks all bad disk sectors.

- 4) The system Linker automatically links separately compiled “Units” of p-code. This allows you to build up libraries of useful procedures which can then be used by many different programs.

- 5) The interpreter automatically converts the p-code to executable code.

The system is relatively easy to learn. Because it is highly integrated, it is very easy to use. Although expensive, it is useful for beginners, as well as professional programmers. There is also a full range of application programs available which run under the p-System, including spreadsheets, word processors, accounting programs and a host of others. Several well known programs were developed using the p-System. These include VisiSchedule from VisiCorp and MBA from Context.

There is an alternate version of the p-System which offers hard disk support, twenty-five percent more storage, and a number of other utilities. Updates are available, and excellent technical support is provided by the company. Network Consulting, Inc.

UHF Ultra High Frequencies.

Unibus A mini-computer bus with more than 100 signals invented by DEC for its PDP-11. It is not used by the LSI-11. (Trademark of Digital Equipment Corp.)

UNIX A mini- and microcomputer operating system developed by Bell Labs which features multi-programming, a hierarchical file structure, and numerous useful utilities. See XENIX, C—Programming Language.

Unstack To remove from the top of a stack. See POP.

uP Microprocessor.

Up Arrow See Arrow Up.

UPC Universal Product Code.

UPI Universal Peripheral Interface.

Upload a Communications File The upload feature in Telcom terminal mode allows you to send a previously created RAM file to the host computer. For instance, you could use TEXT to write a report

while you are traveling. When the report is complete, you can call the computer at your home office and upload the file. Press F3, the Up key, and when you see the prompt "file to upload?" on the screen enter the file name you want to send, then press ENTER. Telcom responds by displaying the prompt width? You then enter a line width between 10 and 132 characters per line and press ENTER. The default width is 40 characters per line. Telcom inserts carriage returns, formats the "page" width, and prevents split words. In some cases this is essential; Compuserve cannot handle data input at more than 132 characters without a carriage return. While uploading, the label UP appears in reverse video. If you press any key during an upload, that character will be inserted into the file you are sending.

Upward Compatible Indicates that programs developed for one version of a programming language, operating system, application software package, or computer will work without alteration on an expanded version of the same language, system, or package. In hardware, upward compatibility refers to the possibility of expanding the power of a system without reprogramming.

us Microsecond or one millionth of a second.

USASCII-8 See ASCII.

USART Universal Synchronous/Asynchronous Receiver/Transmitter. A chip handling all the operations associated with synchronous data communications, such as bisync.

USASCII See ASCII.

User A User is a person who owns or uses a computer.

User Friendly Often, the programs you write will be for your own use. But if you intend others to use your program, you need to give some thought to making your program easy to use, as well as technically correct. Give clear prompts for every item of input data, freeze the screen long enough for them to read or act on the information displayed, provide clear error messages if anything is entered incorrectly or error situations arise, etc. This is often referred to as "human engineering."

Several other guidelines exist for writing user friendly programs. If a complex series of data items has been typed in, and some of those items turn out to be invalid, the user should only have to reenter the bad items. Error messages should indicate which entry is invalid, in what way it is invalid and, if possible, some hints to correct it (such as a list of possible correct values).

User Group A group or club focused on an aspect of computers. Some clubs direct their attention to one particular computer. Other groups focus on a language (FORTH User's Group, Pascal User's Group, etc.), on an Operating System (CP/M, etc.), on an area of application (accounting, education, science, or graphics.) Many magazines on computing carry lists of clubs/groups and report on their activities. You may also be interested in on-line user groups offered by some of the information services. Compuserve, for instance, offers several on-line special interest groups, including one for the Model 100.

User groups provide a valuable opportunity to get and give advice on PC hardware, software, and applications. Often you can talk to someone who used a product you are considering buying. User group newsletters may also offer useful information. If you need a programmer or consultant, you may meet or hear about a good one at a User Group meeting. See Compuserve.

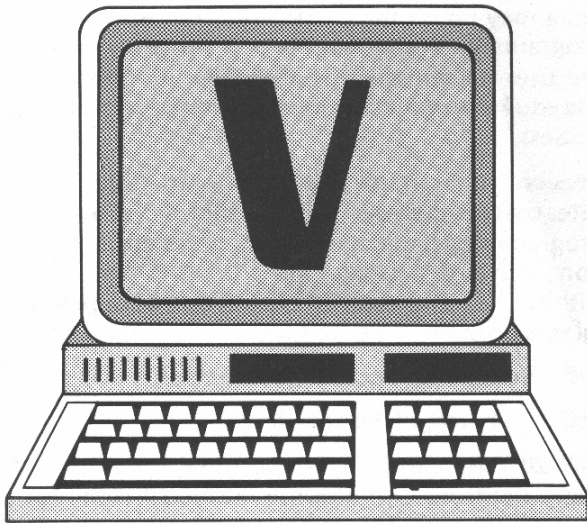
USRT Universal Synchronous Receiver/Transmitter. A serial-to-parallel converter for high-speed communications.

Utilities Software used for routine computer tasks. They aid the operation and use of the computer for a number of different applications and uses. Examples of utilities are an editor, a sorter, a debugger, or a file handler.

Utility Packages See COPY; DATA+; GRAPH+; PortaDex; PortaMax; PortaMed; PortaPrint; SIZE; SORT; SORT+; and SORT2+.

UUT Unit Under Test.

UV Ultraviolet.



V Codes. ASCII 86, HEX 56. v—ASCII 118, HEX 76.

V Volt. Also, the overflow status flag.

VAL BASIC Function. Returns the numerical value of the beginning of string <x\$>. The format is:

<v> = VAL(<x\$>)

<x\$> is a string expression.

The VAL function strips blanks, tabs, and line feeds from the argument string and returns only the numeric characters that preface the string. If the first characters of <x\$> are not numeric, then VAL(<x\$>) will return 0. See STR\$. Here is an example of this function:

```
PRINT VAL ("52 CASES")
52
OK
```

Variable A symbolically named entity which may assume an assigned value, or a number of values. Variable types in Model 100 BASIC are single- and double-precision numeric, integer, and string.

Variable Names Rules for Variables in BASIC.

- a) must start with a letter.
- b) can have any number of characters, but only the first two are significant. BASIC would not be able to differentiate between variables named SUN and SURPRISE for example.
- c) cannot be a reserved word such as IF, ON, THEN, GOTO, etc., or a reserved word followed by a type declaration character (\$, %, ?, !, #), or a reserved word embedded in a variable name, as END is embedded in MEND. See BASIC Reserved Words.
- d) the default variable type in BASIC is a double-precision numeral unless the variable name ends with one of the following type declaration characters:

\$ is the type declaration character for a string variable (0 to 255 letters, numbers, punctuation marks, and other characters);

% is the type declaration character for numeric integers (whole numbers from -32768 to +32767);

! is the type declaration character for single-precision numbers (numbers with decimal fractions and up to six significant digits);

is the type declaration character for double-precision numbers (numbers with decimal fractions and up to fourteen significant digits).

Remember, the default value for the type of variable is double-precision, so any variable name not ending in \$, !, or % is automatically a double-precision numeric variable. An exception is any variable which begins with a character specified in one of the type declaration statements:

DEFINT—for integer numbers

DEFSNG—for single-precision numbers

DEFDBL—for double-precision numbers

DEFSTR—for string variables

For example, if you place the statement:

DEFSTR A,B,C

at the beginning of your program, all variables starting with A, B, or C (such as AXE, BOX, or CAT) are string variables because they start with one of the characters defined in the DEFSTR to be a string variable prefix.

Type declaration characters added to a variable name override any variable type definitions resulting from a DEFINT, DEFSNG, DEFDBL, or DEFSTR statement. For example, if you used DEFSTR to define variables starting with A as string variables, the variable AT% would still require integer data, resisting definition as a string variable. See Type Declaration Character and Type Declaration Statement.

Variable Types, Numeric See Numeric Variable.

Variable Types, Numeric—Converting See Numeric Variables, Converting Between Types.

VARPTR BASIC Function. Returns the address in memory of a BASIC variable. The format is:

v = VARPTR(<variable>)

<variable> is the name of a numeric or string variable or array element in your program. You must assign a value to <variable> prior to the call to VARPTR or an "?FM Error", an illegal function call error, results.

The address returned is an integer value expressed in decimal format. Addresses over 32737 return negative values. The VARPTR function returns the

address of the first byte of data identified with <variable>. All simple variables should be assigned before calling VARPTR for an array since addresses of arrays change whenever a new simple variable is assigned.

VARPTR is generally used to retrieve the address of a variable or array so it may be passed to a Machine language subroutine. A function call of the form VARPTR(A(0)) is usually specified when passing an array, so that the lowest-addressed element of the array is returned.

VAX A 32-bit minicomputer manufactured by Digital Equipment which can execute PDP-11 Machine language.

VDI Video Display Input.

VDI Video Display Terminal. The term used in the newspaper community for CRT.

VDU Video Display Unit. The British term for CRT.

Vector Display A CRT which moves the electron beam randomly to trace figures on the screen, whereas a raster display sweeps the beam through a fixed pattern, building up an image with a matrix of points. Vector displays are used in many arcade games.

Vectored Interrupt Interrupt scheme where information about the event causing the interrupt is provided by hardware at the time of the interrupt.

Vectored Interrupt An interrupt which carries its identity number, or the address of its handler.

Vectoring Automatic branching to a specified address. See Interrupt.

Verify Cassette Load After you have loaded a program to BASIC from cassette, you can use the CLOAD? function to compare the cassette file with the copy in BASIC to verify that it was loaded correctly. See CLOAD?.

Version Programs and software packages are updated to correct errors or add new capabilities. To keep programs from being in a constant state of flux and to simplify distributing modified programs to users, a number of changes are made at once, tested, and packaged as a new version or release of the program.

The first version or release is typically called Version 1.0. The first minor revision is called 1.1, the second minor revision, 1.2, etc. When a major change or large number of minor changes have been made, a new number may be assigned, as Version 2.0. Since more than one version of a pro-

gram may be on the market, version numbers let programmers know with which program version the user has encountered problems. Numbering lets you know which changes are in the version you possess.

Versions of Programs, Upward Compatible Indicates that programs developed for one version of a programming language, operating system, application software package, or computer will work without alterations on an expanded, more powerful version of the same language, system, or package.

V/F Voltage to Frequency converter.

VHF Very High Frequency.

VIC 20 An inexpensive, personal color-output computer from Commodore Business Machines.

Video Signal An electronic signal containing specifications for the location and brightness of each point on a CRT screen, and timing signals to place the image properly on the screen.

Viking* A simulation game. 24K. Chattanooga Choo Choo Software (for Prickly-Pear Software).

VIP An RCA board using the COSMAC MPU.

Virtual Address A user- or system-generated address which references objects in a logical address space regardless of their physical memory location. A virtual address must be translated by the operating system into a valid physical address which may, in turn, involve the movement of data between primary and secondary storage (usually disk).

Virtual Memory The memory address space available to any process running on the processor. It may be larger than the physical memory.

VisiCalc* The original electronic spreadsheet. VisiCalc translates once complicated programming procedures into a form and procedure resembling those which businessmen commonly carry out on paper. The program also provides sufficient processing sophistication to handle many business planning and forecasting needs.

On screen, VisiCalc emulates a large piece of paper, similar to a financial worksheet on which statistics are recorded. The screen is divided into a matrix of columns and rows, which intersect to form small rectangular cells, identified by row and column coordinates. Numbers and formulas can be placed in these cells.

VisiCalc remembers how these numerical values are derived and can recalculate a great web of interrelated statistics based on one change. VisiCorp.

VLSI Very Large Scale Integration. A technology permitting the use of over 10,000 transistors per chip.

VMOS Vertical MOS. Cutting a V-shaped groove in the silicon substrate to increase the density of components per square inch.

Volatile Storage Storage which erases its contents when power is removed.

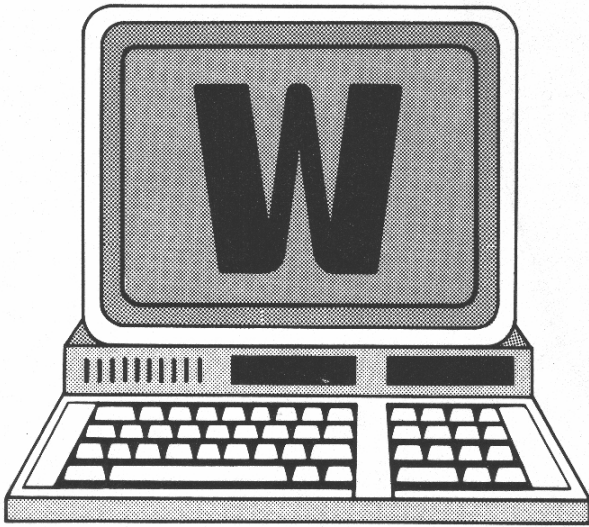
VOM Volt Ohm Milliammeter. A test instrument that measures voltage, resistance, and current. It is usually portable, with an analog meter or digital display for readout.

VSS Voltage for Substrate and Sources. The ground for MOS circuits.

VSNC Vertical SYNC. A signal in a TV which determines the vertical position of the image.

VTAM Virtual Teleprocessing Access Method (IBM mainframe telecommunications system).

VTR Video Tape Recorder.



W Codes. ASCII 87, HEX 57. w—ASCII 119, HEX 77.

W Write

Wafer A slice of a silicon on which integrated circuits are made. After testing, the wafer is cut up into individual circuits called dice or chips. The dice that were not rejected in the test are packaged and retested before being used as finished integrated circuit components.

Wait State A micro-cycle or internal state entered by an MPU when a synchronizing signal is not present. It synchronizes a fast processor with a slower memory.

Wand A stick which reads optically coded product labels (usually bar codes) found on retail sales items. The Model 100 has an I/O port for interfacing a Hewlett-Packard HEDS-300 compatible bar code reader wand to the computer.

Wand * Supports use of the Hewlett-Packard bar code reader with the Model 100. Portable Computer Support Group.

Way One of several electrical connectors printed on the connecting edge of a printed circuit board (or "card").

WD Western Digital Corp. A manufacturer of processor and controller chips.

WE Write Enable. A control signal which writes to a memory device. Typically used to provide write-protection as in diskette and cassette drives. Also used in bank switch organization.

WEMA Western Electronics Manufacturers Association.

Winchester Disk A hard disk system with very light read/write heads, low head-to-disk clearance, and complete enclosure of the magnetic media in a dust-free environment to achieve high information density and fast access-time. Hard disk and Winchester disk are essentially synonyms.

Window A section of a computer screen which displays specific types of information. See Split Screen.

Wire Wrap Mechanically connecting wires in complex circuits. To make an electrical connection, each wire is tightly wound several turns around a square post. This is seldom used now except for hardware during system development.

WOM Write Only Memory. A semi-humorous term for unusable space in the memory of a computer. Get it? See, you can write things to it, but it won't listen, and it certainly won't let you read them again.

Word A logical unit of information which can have any number of bits. For MPUs, however, a word is usually 4, 8, 16, or 32 bits. For the Model 100, a word is 1 byte or 8 bits.

Word Processor A computer-based system for writing, editing, and formatting documents such as letters, reports, and books. It may be either a specialized hardware system dedicated to these tasks or a program package run on a general purpose computer. See TEXT.

Words, Reserved See BASIC Reserved Words.

WordStar* One of the most widely used micro computer word processing programs in the world. Only the Wang dedicated word processing machines have a comparable number of users.

WordStar's popularity is a tribute to timing and programming. Introduced after the Electric Pencil (the first well-known microcomputer word processor), WordStar was the only program that avoided some of Electric Pencil's worst mistakes. Its file handling capabilities are noteworthy. It saves all of your document, and creates a backup file of the last version entered. WordStar provides a reference for evaluating the TEXT feature of the Model 100.

Word/Text Processors See PortaPrint, PRINT, and WRITE+.

Workspace An area of memory allocated for working storage.

WPM Words Per Minute.

WRITE+* • WV

WRITE+* A very powerful word processor for use in many professions. You can create letter formats, documentation, graphs, title pages, outlines, and text.

When combined with the W+SPEC and CORTNS programs, additional commands are available to enhance formatting abilities. Line spacing, page control, paragraph indentation, headings, footers, underlining, and searching are a few of the special features offered. This is part of BusinessPak+. 16K; cassette. Portable Computer Support Group.

Write Data to File BASIC. See PRINT#, PRINT# USING, SAVE, and SAVEM.

Write-Protect Prevent information from being written onto a storage medium. Floppy disk jackets often have adhesive tabs which can be placed on the disk to write-protect it by temporarily disabling the disk drive's write circuitry. Cassettes have a plastic break-out tab. See Adhesive Tab and Write-Protected Diskette.

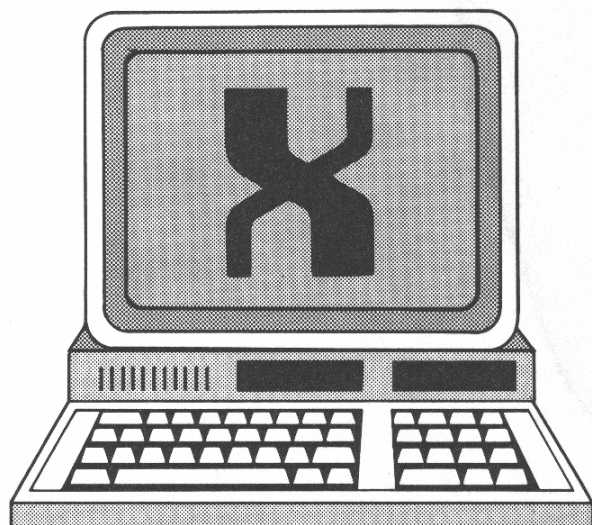
Write-Protected Diskette A write-protected diskette does not have an accessible notch about one inch down on the right side. An adhesive tab may block a small, spring-loaded switch or light beam inside the diskette which temporarily disables the DOS diskette driver programs. An error message appears ("Attempted write-protect violation") any time you attempt to format, change, delete, or copy a file on the write-protected diskette.

You are allowed to use, load, or copy files from the write-protected diskette in case you accidentally lose your only copy of a program. In most cases, you copy the write-protected diskette onto a disk with a notch, store the write-protected diskette as a permanent copy, and use the notched diskette as your working copy.

It is a good practice to put an adhesive tab (supplied with boxes of diskettes) over the write-protect notch of any important diskette you have a backup of. Then, if you accidentally ask for the backup in the wrong direction (from the old diskette to your important diskette), you get a second chance to make the backup rather than losing your data.

WS WorkSpace; or WordStar; or WoodShed; or WhatSoever; or William Shatner.

WV Working Voltage.



X Codes. ASCII 88, HEX 58. x—ASCII 120, HEX 78.

X IndeX Register.

XCHG 80C85 Assembly Language Instruction. The contents of registers H and L are exchanged with the contents of registers D and E. The addressing mode is register. No flags are set.

XENIX The Microsoft implementation of the UNIX operating system for microcomputers.

XMIT TransMIT.

XMT TransMiT.

XR EXternal Reset.

XRA M 80C85 Assembly Language Instruction. Exclusive OR Memory. The content of the memory location whose address is in the H and L registers is exclusive OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared. The addressing mode is register indirect. Z, S, P, CY, and AC flags are set.

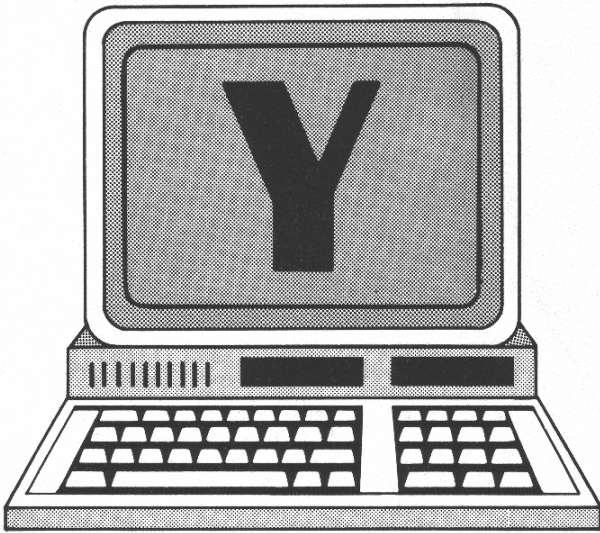
XRA <r> 80C85 Assembly Language Instruction. Exclusive OR Register. The content of register <r> is exclusive OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared. The addressing mode is register. Z, S, P, CY, and AC flags are set.

XRI <data> 80C85 Assembly Language Instruction. Exclusive OR immediate. The content of the second byte of the instruction is exclusive OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared. The addressing mode is immediate. Z, S, P, CY, and AC flags are set.

XTAL CrysTAL.

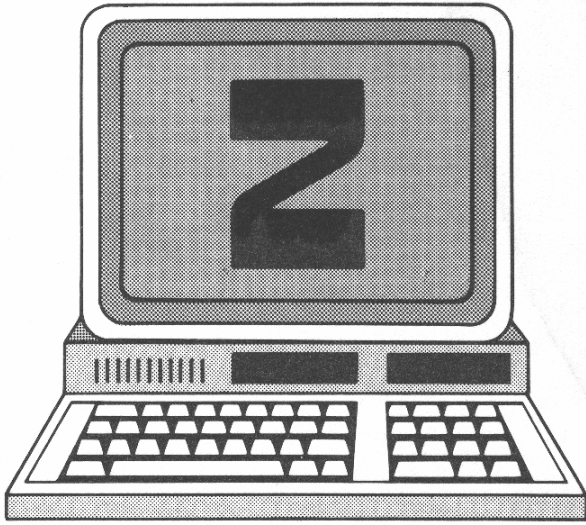
XTHL 80C85 Assembly Language Instruction. The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP. The addressing mode is register indirect. No flags are set.

X-Y Plotter A device which draws points or lines on a sheet of paper based on X and Y coordinates from a computer.



Y Codes. ASCII 89, HEX 59. y—ASCII 121, HEX 79.

Yield The proportion or operational chips in a production batch. $\text{Yield} = \text{good chips} / \text{total chips on wafer}$.

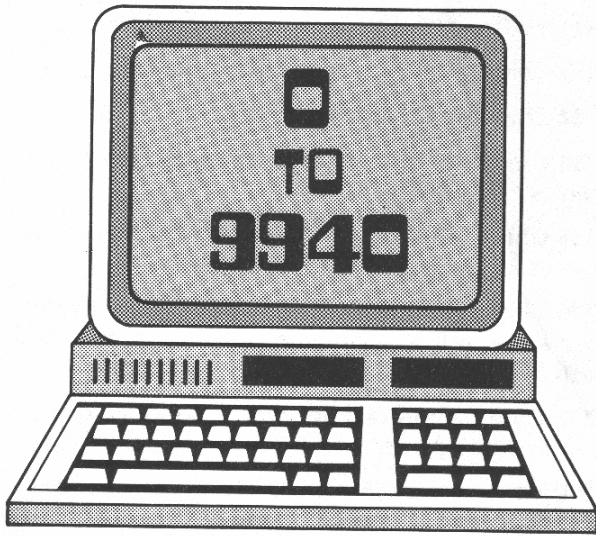


Z Codes. ASCII 90, HEX 5A. z—ASCII 122, HEX 7A.

Z Impedance measured in ohms. Also, the Zero flag.

Zilog Manufacturer of the Z80, and many other widely used chips. Zilog also publishes technical reference material on the Z80 and other microprocessors.

Zones Each group of fourteen spaces across the print line is called a print zone. A comma (,) in an LPRINT list of items to be printed means “start printing the following item at the start of the next print zone.” Contrast this with the semicolon (;), which means the next item is to print immediately after this one, without even a single space between. The print zones begin in columns 1, 14, 28, 42, 56, and 70. See also Print Zones.



0 Used to set the line spacing to 8 lines-per-inch. Enter BASIC statement:

```
LPRINT CHR$(27);"0";
```

See Type Formats.

1 Used to set the line spacing to 72/7 lines-per-inch. Enter the BASIC statement:

```
LPRINT CHR$(27);"1";
```

This is a good setting for spacing with compressed print. See Type Formats.

2 Used to set the line spacing to 6 lines-per-inch. Enter the BASIC statement:

```
LPRINT CHR$(27);"2";
```

See Type Formats.

8 ASCII, Uses of To move the cursor one space to the left, enter the BASIC statement:

```
PRINT CHR$(8);
```

This erases data on the screen as the cursor moves to the left. From the keyboard enter: ←.

8K Memory Models* Available for the TRS-80 Model 100. Easy to install and includes a warranty. Hardware Accessory. Iota Systems.

8K Ram Memory Modules* Memory modules to expand Ram memory in 8K increments up to 32K. BT Enterprises.

8K RAM Memory Modules* Expansion RAM memory modules for the Model 100 plug into existing sockets with no modifications. Holmes Engineering.

8K Ram Memory Modules* Instructions are included for installing these exact replacement expansion RAM modules for the Model 100. Warranty included. PG Design Electronics.

8K Ram Memory Modules* Easy-to-install expansion RAM memory modules for the Model 100. Money back within 90 days of purchase if not satisfied. These folks also offer a printer cable and modem cable for the Model 100 at a reduced price with the purchase of a memory module. Purple Computing.

9-Bit Rotation When the carry bit of the CPU is considered a ninth, high-order bit for the 8-bit register being rotated, it is called 9-bit rotation.

10 ASCII, Uses of To advance one line on the printer (space up) without carriage return, enter the BASIC statement:

```
LPRINT CHR$(10);
```

or use the "line feed" (LF) button on your printer. Entering just LPRINT gives a line feed—both a space up one line and a return to left margin (carriage return).

10 ASCII, Uses of To move the cursor one space down, enter the BASIC statement:

```
PRINT CHR$(10);
```

This only moves the cursor, it does not erase data from the screen. From the keyboard, enter ↓.

12 ASCII, Uses of To advance the paper to the top of a page in the printer, enter the BASIC statement:

```
LPRINT CHR$(12);
```

or use the "top of form" or "form feed" manual control button (FF) on the printer.

You may need to adjust the paper in the printer so it actually is at the top of a page as defined by the perforations. In a program, you may want to provide instructions to the operator and a pause to allow for adjustment of the paper.

27 ASCII, Uses of Enter ASCII 27 (escape) for printer control in LPRINT statements to set lines-per-inch, page length, and print size, or use $128 + 27 = 155$ to avoid complications in some programs. See Type Formats.

32 ASCII, Uses of To move the cursor one space to the right, enter the BASIC statement:

```
PRINT CHR$(32);
```

This erases data on the screen as the cursor moves to the right. From the keyboard, enter →.

48 TPI See 48 Tracks-Per-Inch.

48 Tracks-Per-Inch A disk format called double-density. This means that the tracks on the disk are twice as close together (48 TPI) as on single-density diskettes. Double-sided drives also have two read/write heads, so information is stored on

96 Tracks-Per-Inch • 3861 Chip

both the top and bottom side of the diskette. See Disk Formats, Disk Drive, and 96 Tracks-Per-Inch.

96 Tracks-Per-Inch Also called a quad-drive, this drive writes the tracks twice as close together as a double-density drive. This puts twice as much data within the travel distance of the drive arm. Quad-drives require special software. By also using the 10 sector/track format, the 640K per double-sided quad-diskette can be further increased to 800K per diskette. See 48 Tracks-Per-Inch.

371 Chip Cassette controller. Made by NEC.

372 Chip FDC. Made by NEC.

400 Chip 4-bit I2L slice. Made by TI.

400K Using ten sector-per-track format allows you to store 200K on your single-sided, double-density drives (otherwise 160K) and 400K on your double-sided, double-density drives (otherwise 320K). See 48 Track-Per-Inch and Diskette Formats.

481 Chip 4-bit slice. Made by TI.

601 Chip 16-bit chip. Made by Data General.

1000 Chip 4-bit microprocessor available in the versions listed below. Also called TMS-1000. Made by TI.

1070 See 1000.

1100 See 1000.

1200 See 1000.

1270 See 1000.

1300 See 1000.

1600 Chip Set designed for PDP-11/03 emulation. Made by Western Digital.

1702 Chip An ultraviolet erasable PROM organized as 256 words by 8 bits.

1771 Chip Single-density floppy disk controller chip.

1791 Chip Double-density floppy disk controller chip.

1802 Chip Cosmac 8-bit CMOS microprocessor. Made by RCA.

2102 Chip Common static RAM integrated circuit, organized as 1K by 1 bit.

2114 Chip Static RAM organized as 1K by 4 bits.

2650 Chip 8-bit microprocessor. Made by Signetics.

2651 Chip 2650 UART. Made by Signetics.

2652 Chip SDLC chip. Made by Signetics.

2655 Chip 2650 PIO. Made by Signetics.

2702 Chip See 1702.

2708 Chip An ultraviolet-erasable PROM organized as 1K by 8 bits.

2716 Chip An ultraviolet-erasable PROM organized as 2K by 16 bits. Made by Intel.

2716 Chip An ultraviolet-erasable PROM organized as 2K by 8 bits. Not compatible with the Intel part. Made by TI.

2732 Chip An ultraviolet-erasable PROM organized as 4K by 8 bits.

2900 Chip A family of 4-bit slice components. Widely used to construct special-purpose controllers and microprocessors. Introduced by AMD and second-sourced by many other manufacturers.

2901 Chip A 4-bit slice processor. Made by AMD.

2902 Chip Look-ahead carry generator. Made by AMD.

2903 Chip An improved version of 2901. Made by AMD.

2909 Chip Microprogram sequencer. Made by AMD.

2911 Chip Microprogram sequencer. Made by AMD.

2914 Chip PIC. Made by AMD.

3000 Chip Family of 2-bit slice components. Made by Intel.

3001 Chip Microprogram control unit. Made by Intel.

3002 Chip Central processing element 2-bit slice. Made by Intel.

3003 Chip Look-ahead carry generator. Made by Intel.

3850 Chip The F8 family processor chip. Part of an 8-bit, 2-chip microcomputer. Made by Fairchild.

3851 Chip The F8 family program storage unit used with the 3850. Made by Fairchild.

3852 Chip Dynamic memory interface for the F8. Made by Fairchild.

3853 Chip SMI for the F8. Made by Fairchild.

3854 Chip DMA for the F8. Made by Fairchild.

3861 Chip PIO for the F8. Made by Fairchild.

3870 Chip An 8-bit, one-chip microcomputer. Contains 4032 bytes of ROM, and 128 bytes of RAM. Made by Mostek.

3876 Chip An 8-bit, one-chip microcomputer. The upgrade of the 3870 which contains 4032 bytes of ROM, and 256 bytes of RAM. Made by Mostek.

3880 Chip Mostek Z80.

4004 Chip 4-bit microprocessor. Made by Intel.

4040 Chip 4-bit microprocessor. The upgrade of the 4004 which contains more registers and executes a larger instruction set. Made by Intel.

4044 Chip Static RAM organized as 4K by 1 bit.

4116 Chip Dynamic RAM organized as 16K by 1 bit.

4164 Chip Dynamic RAM organized as 64K by 1 bit.

4264 Chip 4040 PIO.

4308 Chip ROM 1K by 8 with I/O ports for the 4040. Made By Intel.

5701 Chip MMI 4-bit slice predecessor of the 2901 mil version.

6100 Chip Intersil 12-bit CMOS microprocessor which emulates the PDP-8.

6502 Chip 8-bit microprocessor. Widely used in mass-marketed computer systems such as Apple, Pet, and Atari. Made by MOS Technology.

6520 Chip PIO. Made by MOS Technology.

6530 Chip RAM, ROM, I/O, and timer. Made by MOS Technology.

65XX Chip Support chips belonging to the 6502 family. Made by MOS.

6701 Chip Same as the 5701 in the commercial version.

6800 Chip 8-bit microprocessor. Made by Motorola.

6801 Chip 8-bit, 1-chip microcomputer.

6802 Chip 8-bit, 2-chip microcomputer. Upgrade of the 6800, that contains functions previously in the other 6800 family components. Made by Motorola.

6809 Chip 8-bit, high performance upgrade of the 6800. Has an expanded instruction set and 16-bit word handling capability. Made by Motorola.

6820 Chip 6800 PIO. Made by Motorola, Fairchild, and Mostek.

6828 Chip PIC. Made by Motorola.

6845 Chip CRT controller. Made by Motorola.

6850 Chip 6800 UART. Made by Motorola.

6860 Chip Modem. Made by Motorola, Fairchild and AMD.

6870 Chip Clock. Made by Motorola.

7400 Chip Series of TTL logic. Made by TI.

8008 Chip 8-bit microprocessor. Made by Intel.

8048 Chip 8-bit family of 1-chip microcomputers with 1-chip RAM and ROM. The 8748 version has an EPROM on the same chip as the processor.

8080 Chip 8-bit microprocessor. Upgrade from the 8008, which has a different instruction set but retains a similar architecture to the 8008. The 8008 was the dominant microprocessor of the 1970s. Because it runs a large library of CP/M code, the 8080 is available for the PC or the XEDEx Baby Blue Card. Made by Intel.

8085 Chip 8-bit microprocessor. Upgrade of the 8080, contains functions that were previously on other 8080 family chips, as well as two extra instructions and four interrupt levels. Made by Intel.

80C85 Chip 8-bit microprocessor. A CMOS version of the 8085 chip used as the Model 100 CPU.

8086 Full 16-bit version of the 8088.

8086 Chip 16-bit, byte-oriented microprocessor that resembles the 8085, but has an expanded instruction set and 16-bit arithmetic capabilities. Made by Intel.

8087 Chip Numeric data co-processor for the 8086 and the 8088. Implements proposed IEEE floating-point standard.

8088 16-bit processor from Intel used as the CPU for the PC. The 8088 is a slightly stripped-down version of Intel's original 1978 8086. The 8088 differs from the 8086 in the following ways:

1) The 8088 uses an 8-bit data bus, as opposed to the 16-bit data bus of the 8086. The 8088 needs about twice as many memory cycles to fetch the same amount of data as the 8086.

2) The instruction "fetch-ahead" queue of the 8088 is 4 bytes, versus 6 bytes in the 8086. The 8088 has to wait for instructions to be fetched from memory more often than the 8086.

The 8088 is thus a 16-bit processor internally, but with an 8-bit bus structure. Other than this, the 8088 and 8086 are the same. Even at the Assembly

8089 Chip • 9940 Chip

language and Machine Code level, they are interchangeable to the programmer.

The 8086/8088 processors have built-in provisions for being paired with a co-processor which can perform certain functions faster than the main processor. For example, the 8087 is specialized for floating point operations. When working with the 8087, the 8088 passes all floating point arithmetic to the 8087 and waits for it to signal completion. The result is execution up to 100 times faster for programs with extensive floating point calculations.

The 8086/8088 processor can assemble programs for the earlier 8008 and 8080 processors, but the object codes are not compatible.

8089 Chip 16-bit input/output processor. Made by Intel.

8155 I/O Chip An 8085 family member with 256 8-bit bytes of memory and six registers.

8155 I/O Chip This is the input/output chip of the Model 100. It is in the 8085 family of chips and features 256 bytes (eight bits each) of memory, as well as six registers. This makes the 8155 chip highly programmable—although the Model 100 probably does not make full use of the onboard memory. Interfacing the memory on the chip to the rest of the computer would use up about 2K of address just for 256 bytes of memory space. If any of the onboard memory is used, it is paged in and out.

The 8155 I/O chip is mapped to eight continuous ports at all times. The Model 100 uses only six of these. On the port address bus the first five bits address the 8155 chip as a whole. The lower three bits select between the six registers on the 8155 chip. The lowest register is the command/status register. The next two are ports A and B, which are 8-bit I/O ports. Ports A and B are programmable for input or output. The next register is a 6-bit port, port C. Port C has several uses. It can be programmed for input, for output, or to handle the handshaking for ports A and B. The next two ports constitute a 14-bit timer. When the Model 100 is initialized, ports A and B are set to output and port C to input.

8212 Chip Parallel latch and buffer in the 8080 family. Made by Intel.

8224 Chip Clock generator for the 8080. Made by Intel.

8228 Chip System controller for the 8080. Made by Intel.

8251 Chip USART for the 8080 family. Also called a PCI. Made by Intel.

8253 Chip Programmable interval timer for the 8080 family. Made by Intel.

8255 Chip Programmable parallel interface for the 8080 family. Made by Intel.

8257 Chip Direct memory access controller for the 8080 family. Made by Intel.

8259 Chip Interrupt controller for the 8080 family. Made by Intel.

8271 Chip Single-density floppy disk controller in the 8080 family. Made by Intel.

8273 Chip Synchronous data link controller in the 8080 family. Made by Intel.

8275 Chip CRT controller in the 8080 family. Made by Intel.

8279 Chip Keyboard and display controller in the 8080 family. Made by Intel.

8291 Chip IEEE 488 bus talker/listener interface chip. Made by Intel.

8292 Chip IEEE 488 bus controller chip. Made by Intel.

8708 Chip See 2708 Chip.

8748 Chip 8048 with EPROM on the same chip as the processor. Made by Intel.

9080 Chip AMD's 8080.

9400 Chip Bipolar Macrologic family. Made by Fairchild.

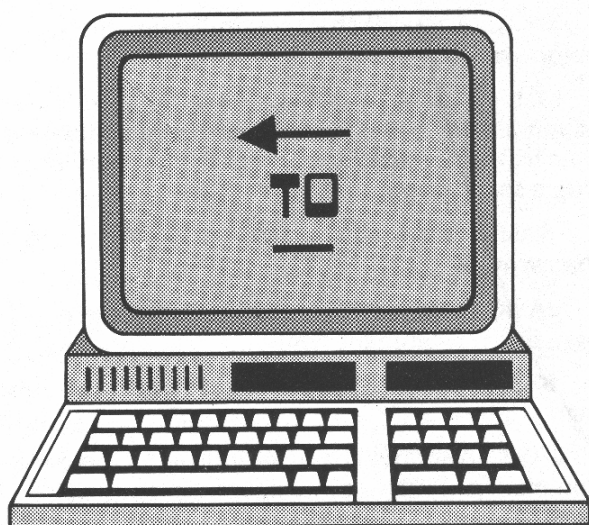
9511 Chip Arithmetic processing chip. Made by AMD.

9900 Chip 16-bit microprocessor compatible with the 990 series of minicomputers. Made by TI.

9904 Chip 9900 clock. Made by TI.

9914 Chip IEEE 488 bus interface chip. Supports talker/listener and controller functions.

9940 Chip 16-bit, one-chip microcomputer.



← The left arrow backspaces, or moves the cursor left one space. Holding down the key repeats the move. In all modes except TEXT and BASIC EDIT, backspacing erases the characters the cursor presses over. In TEXT, and in TEXT as invoked by the BASIC EDIT, ← performs additional functions if used with the SHIFT and CTRL keys. Pressing SHIFT and ← moves the cursor to the first character of the current word, or, if it is already there, to the first character of the next word. Pressing CTRL and ← moves the cursor to the first character of the current line. See Cursor.

→ The right arrow moves the cursor one space to the right. Holding down the key repeats the move. In TEXT and in TEXT as invoked by the BASIC EDIT, → performs additional functions if used with the SHIFT or CTRL key. Pressing SHIFT and → moves the cursor to the first character of the next word to the right. Pressing CTRL and → moves the cursor to the rightmost character on the current line. See Cursor.

↓ In TEXT and BASIC EDIT, the cursor down key moves the cursor down one line. Holding down the cursor down key causes this process to autorepeat. In TEXT, and in TEXT as invoked by the BASIC EDIT, ↓ performs additional functions if used with the SHIFT or CTRL key. Pressing SHIFT and ↓ advances the cursor to the last line, same column of the LCD, or, if the cursor is already there, the next seven file lines scroll onto the LCD. CTRL and ↓ takes you to the last eight lines of the file. See Cursor.

↑ In TEXT and BASIC EDIT, pressing the cursor up key moves the cursor up one line. Holding down the cursor up key causes this process to autorepeat. In TEXT, and in TEXT as invoked by the BASIC EDIT, ↑ performs additional functions if used with the

SHIFT or CTRL key. Pressing SHIFT and ↑ moves the cursor to the first line of the LCD; if it is already there, the previous seven lines of the file scroll onto the LCD. See Cursor.

“ BASIC. A special character indicating the double quotation mark or string delimiter.

” To get one or more spaces between fields printed by your BASIC programs, use a literal of spaces like: “ ” To get several spaces between the printed values of A\$ and B\$ use:

LPRINT A\$;“ ”;B\$

See Print Zones.

BASIC. A special character indicating the number (or pound) symbol, or the double-precision type declaration character. See Type Declaration Characters.

Double-precision numbers (numbers with decimal fractions up to fourteen significant digits) can be declared by having their variable names end in #. See BASIC Variable Names.

\$ BASIC. A special character indicating the dollar sign, or the string type declaration character. See Type Declaration Characters.

\$ BASIC. String variable names in BASIC must end in \$ or start with a series of letters specified in a DEFSTR statement. See BASIC Variable Names.

% BASIC. A special character indicating the percent symbol or the integer type declaration character.

% BASIC. Numeric variable type delimiter. See BASIC Variable Names.

' BASIC. A special character, indicating the single quotation mark, apostrophe, or remark delimiter.

(BASIC. A special character, indicating the left parenthesis.

) BASIC. A special character, indicating the right parenthesis.

* BASIC. A special character, indicating the times sign (multiplication).

* RAM files listed on the Main Menu or by the FILES BASIC command may be followed by the * if that file is the BASIC file currently in BASIC memory.

+ BASIC. A special character, indicating the plus sign (addition).

, BASIC. A special character, indicating the comma delimiter.

, Each group of 14 spaces across the print line is called a print zone. A comma in an LPRINT list of items to be printed means "start printing the following item at the start of the next print zone." See Print Zones.

,, To leave space on the print line between items, put an extra comma in the print list. Enter the BASIC statement:

LPRINT A,,B

to print A in print zone 1, nothing in print zone 2, and B in print zone 3. See Print Zones.

- BASIC. A special character indicating the minus sign (subtraction).

. BASIC. A special character indicating the period or decimal point.

. In a file name, the period is used to delimit the file name from the password. For example:

RAM: EXAM.DO

is a RAM file named "exam" stored in ASCII (document format).

,A While using the SAVE or CSAVE command, you can specify ,A at the end of the BASIC command to save the current contents of BASIC memory as an ASC# format file.

.BA The file extension for a RAM file stored in tokenized BASIC format. The operating system automatically assigns this extension to files SAVED from BASIC memory or LOADED to RAM from BASIC cassette files, unless you use the ,A option of the SAVE command.

.CO The file extension for a RAM file stored in Machine language format.

.DO The file extension used for RAM ASCII format files created in text mode or saved from BASIC memory to RAM using the ,A option of the SAVE command.

.P .P is for protected BASIC programs in internal format created with SAVE,P command while using the BLOAD command in direct mode.

/ BASIC. A special character indicating the division symbol or slash.

: BASIC. A special character indicating the colon symbol or statement separator.

: In a BASIC file name, the : is used to separate the drive name from the file name. For example "RAM:TALK.BA" is a file stored in RAM, named "talk," that is stored in BASIC tokenized format.

; BASIC. A special character indicating the semicolon symbol.

; Each group of 14 spaces across the print line is called a print zone. A semicolon means the next item to print is immediately after this one, without a single space between. See Print Line.

< BASIC. A special character indicating the less than symbol.

= BASIC. A special character indicating the equal sign, or an assignment symbol.

> BASIC. A special character indicating the greater than symbol.

? BASIC. A special character indicating the question mark or PRINT abbreviation.

? ? is the prompt from a program written in the BASIC language which is running and needs you to type in data in answer to a question.

\ BASIC. A special character indicating the integer division symbol or backslash.

^ BASIC. A special character indicating the exponentiation symbol or caret.

_ BASIC. A special character indicating the underline symbol.

AMPI

705 N. Bower
Richardson, TX 75081
Phone: 214-238-1815

Bluestem Productions

Box 334
Wayzata, MN 55391
Phone: 612-471-7795

BT Enterprizes

10 B Carlough Rd.
Bohemia, NY 11716
Phone: 800-645-1165

CompuServe

P.O. Box 20212
5000 Arlington Centre Blvd.
Columbus, OH 43220
Phone: 800-848-8199

Dow Jones Information Services

P.O. Box 300
Princeton, NJ 08540
Phone: 800-257-5114

Iota Systems

1690 Day Valley Rd.
Aptos, CA 95003
Phone: 408-684-0482

Micro Computing Services

1637 N. Jantzen
Portland, OR 97217
Phone: 503-285-7424

Portable Computer Support Group

11035 Harry Hines Blvd. #207
Dallas, TX 75229
Phone: 214-351-0564

Purple Computing

4807 Calle Alto
Camarillo, CA 93010
Phone: 805-987-4788

SilverWare

P.O. Box 21101
Santa Barbara, CA 93112
Phone: 805-966-1449

Skyline Marketing Corp.

442 Sunnyside
Wheaton, IL 60187
Phone: 312-286-0762

Source Telecomputing Corp.

1616 Anderson Rd.
McLean, VA 22102
Phone: 702-734-7500

Traveling Software, Inc.

11050 5th Avenue N.E.
Seattle, WA 98125
Phone: 206-367-8090

Tymnet, Inc.

2710 Orchard Pkwy.
San Jose, CA 95134
Phone: 408-946-4900

\$14.95 U.S.A.

THE TRS-80 USER'S ENCYCLOPEDIA (MODEL 100 and NEC PC 8201)

complete ready reference book that is a must for every TRS-80 owner.

This comprehensive book will answer your questions, give you "inside" info, and greatly increase your use and enjoyment of your computer. It covers every aspect of the TRS-80 Computer.

- ▶ Explains programming languages, including BASIC
- ▶ Simplifies operating procedures
- ▶ Guides you through TRSDOS, Assembly language, etc.
- ▶ Describes hundreds of software packages and hardware accessories, with vendor information
- ▶ Lists publications, and other information sources
- ▶ Clarifies general microcomputer terminology and concepts
- ▶ And much more.

EASY TO USE. All entries are concise, easy to understand, and exhaustively referenced to make information instantly accessible.

EASY TO READ. The TRS-80 User's Encyclopedia (Model 100 and NEC PC 8201) covers concepts of interest to the more experienced user, but is written so clearly that even a computer novice can understand it.

ECONOMICAL. Your User's Encyclopedia will pay for itself many times over, thanks to the breadth of its information, its time-saving programming hints, and its thorough product listings.

ESSENTIAL. Your User's Encyclopedia will be the most useful, most frequently consulted book in your computer library.

004-1650321 T864x P554t XT
Phillips, Gary
The TRS-80 user's
encyclopedia (model 100 and
NEC PC-8201)

WAYS, INC.

THE BOOK DIVISION

ISBN 0-912003-13-8